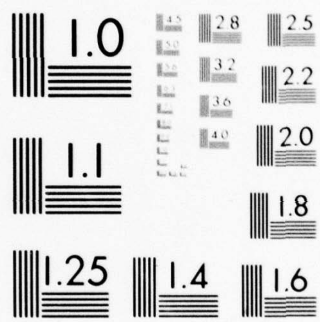
AD-A070 991 ROCKWELL INTERNATIONAL LOS ANGELES CALIF LOS ANGELE--ETC F/G 1/3
MISDEM COMPUTER SIMULATION. VOLUME II. ANALYST MANUAL.(U)
MAY 79 G L GALLIEN, S C SILVER N00123-76-C-0159
UNCLASSIFIED NA-76-372 JTCG/AS-76-S-004 NL

1 OF 2
AD
A070991


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

LEVEL IV

REPORT JTCG/AS-76-S-004

FIELD OF INTEREST: 18.01

(12) B.S.

AD A070991

JTCG/AS

MISDEM COMPUTER SIMULATION (VOLUME II, ANALYST MANUAL)

Final Report

G. L. Gallien
S. C. Silver

May 1979

Approved for public release; distribution unlimited. Statement applied May 1979.

Prepared for

THE JOINT LOGISTICS COMMANDERS
JOINT TECHNICAL COORDINATING GROUP
ON
AIRCRAFT SURVIVABILITY

FOREWORD

Los Angeles Aircraft Division of Rockwell International personnel developed the Mission Damage Effectiveness Model computer simulation under Contract No. 1265R175900 issued by Armament Systems Incorporated, Anaheim, California. The work was performed between April 1975 and August 1976. Marvin Gove, Analysis Branch, NWC (Naval Weapons Center), acted as contract administrator for the documentation of the program under NWC Contract N00123-76-C-0159.

The work was sponsored by the JTCG/AS as part of a 3-year TEAS (Test and Evaluation Aircraft Survivability) program. The TEAS program was funded by DDR&E/ODDT&E. The effort was conducted under the direction of the JTCG/AS Survivability Assessment Subgroup, as part of JTCG/AS Project SA-6-02, *Trade Studies.*

The purpose of the user and analyst manuals is to provide a current documentation of the methodology and easy update maintenance for future program applications on a page-by-page basis.

G. L. Gallien and S. C. Silver, with the program managership of R. L. Moonan, were the key Rockwell International personnel responsible for the computer program development and documentation.

NOTE

This technical report was prepared by the Survivability Assessment Subgroup of the Joint Technical Coordinating Group on Aircraft Survivability in the Joint Logistics Commanders' organization. Because the Services' aircraft survivability development programs are dynamic and changing, this report represents the best data available to the subgroup at this time. It has been coordinated and approved at the JTCG subgroup level. The purpose of the report is to exchange data on all aircraft survivability programs, thereby promoting interservice awareness of the DOD aircraft survivability program under the cognizance of the Joint Logistics Commanders. By careful analysis of the data in this report, personnel with expertise in the aircraft survivability area should be better able to determine technical voids and areas of potential duplication or proliferation.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER JTCG/AS-76-S-004	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) MISDEM Computer Simulation, Volume II. Analyst Manual.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report. Apr 75-Aug 76
		6. PERFORMING ORG. REPORT NUMBER NA76-372
7. AUTHOR(s) (10) G. L. Gallien S. C. Silver		8. CONTRACT OR GRANT NUMBER(s) (15) N00123-76-C-0159
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rockwell International, Los Angeles Aircraft Division 5701 W. Imperial Highway Los Angeles, CA 90009		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS JTCG/AS SA-6-02
11. CONTROLLING OFFICE NAME AND ADDRESS JTCG/AS Central Office Naval Air Systems Command, AIR-5204J Washington, D. C. 20361		12. REPORT DATE (11) May 1979
		13. NUMBER OF PAGES 166
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Weapons Center Systems Development Department (Code 3181) China Lake, CA 93555		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited. Statement applied May 1979.

(12) 178 p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(14) NA-76-372

18. SUPPLEMENTARY NOTES

(18) JTCG/AS (19) 76-S-004

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Survival Probability
Survivability
Mission Effectiveness
Effectiveness Model
Math Model

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See reverse.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 245

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

CONTENTS

Introduction 1

Mathematical Model 5
  Theoretical Overview 5
  Electronics Mode 6
    General 6
    Outputs of the Program 6
    Modes and Equipment 7
    System States 8
    Transition Algorithms 10
    State Generation 23
    Mode Sequence Indentification 24
  Vehicle Mode 27
    Outputs, States, and Mode Logic 27
    Transition Algorithm 35

Schematic Flow Diagrams 41
  Overview 41
  Program Organization 41
  Detailed Flow Diagrams 41

Source Listing 78
  Overview 78
  Simulation Source Deck 79
  Electronics Mode Test Deck 98
  Vehicle Mode Test Deck – Program 1 102
  Vehicle Mode Test Deck – Program 2 104

Simulation Model 106
  Program 1 106
  Program 2 116
    Main Routine 116
    Subroutine TWODNZ 137
  User Information 138
    Machine Requirements 138
    Conversion to CDC Machines 139
    Uncontrolled Errors 139
    Controlled Errors Versus Running Time 140
    Application Notes 141

Figures:

1. Schematic of Mission Effectiveness Analysis 2
2. Mission Damage Effectiveness Model Simulation Summary 3
3. Electronics Mode Data Flow (Example) 9
4. Conventional Damage Transition Algorithm 13
5. Warhead Burst Point Selection 14
6. Offset Zone Probability (Example) 15
7. Nuclear Miss Distance Distribution Model 17
8. Nuclear Transition Algorithm 18
9. Subsystem – Lethal Radius Model 19
10. Compute Lethal Radius 20
11. Assign Vulnerability Index Numbers 22
12. Subsystem Nuclear Kill Model 24
13. Define Mode Sequence and Subsystems Used 28
14. Assign Mode Sequence Numbers 29
15. Schematic Scenario for Vehicle Mode, Eight Events 31
16. Simple Example of Mode Logic for Vehicle Case 32
17. Transition Algorithm for Vehicle Mode 37
18. MISDEM Program Simplified Flow Chart 42
19. Program 1 Organization 43
20. Program 2 Organization 45
21. MISDEM Program 1 – System Description Inputs 49
22. MISDEM Program 1 – Event Description Inputs 50
23. MISDEM Program 1 – Read and Write Subfunctional Flow 51
24. MISDEM Program 1 – Initialize for Mode Sequence Identification 52
25. MISDEM Program 1 – Test for Suppressed Subsystems 53
26. MISDEM Program 1 – Define Mode Sequence and Subsystems Used 54
27. MISDEM Program 1 – Assign Mode Sequence Numbers 55
28. MISDEM Program 1 – Repositon Zeros in State Binary Vector (KW) 56
29. MISDEM Program 1 – Write Mode Sequence and Subsystems Used 57
30. MISDEM Program 2 – System Description Inputs 58
31. MISDEM Program 2 – Event Description Inputs 59
32. MISDEM Program 2 – Compute Lethal Radius 60
33. MISDEM Program 2 – Assign Vulnerability Index Numbers 61
34. MISDEM Program 2 – Compute the Reliability of Every Subsystem and Reshuffle Reliability Order 62
35. MISDEM Program 2 – Generate a Prior State of the System 63
36. MISDEM Program 2 – Test for Suppressed Systems in Prior State and Reorder Subsystems 64
37. MISDEM Program 2 – Generate Current State of the System 65
38. MISDEM Program 2 – Test for Suppressed Subsystems in Current State and Reorder Subsystems 66
39. MISDEM Program 2 – Quick Conventional Damage 67
40. MISDEM Program 2 – Slow Threat Damage 68
41. MISDEM Program 2 – Quick Nuclear Damage and Reliability and Non-Defensive Transitions 69

42. MISDEM Program 2 – Compute State Probabilities and Reset Current State Vector Zeros 70
43. MISDEM Program 2 – Reset Prior State Vector Zeros.................... 71
44. MISDEM Program 2 – Compute and Write Mode Sequence Probabilities...... 72
45. MISDEM Program 2 – Compute Offensive Events Mission Effectiveness...... 73
46. MISDEM Program 2 – Quick Conventional Kills 74
47. MISDEM Program 2 – Slow Kills.. 75
48. MISDEM Program 2 – Quick Nuclear Kills 76
49. MISDEM Program 2 – Linear Interpolation Routine (Subroutine TWODNZ) 77

Tables:

1. Sample System Configuration 25
2. Mode Sequence Logic.. 26
3. Sample Mode Sequences Identified 27
4. Flight Failure Time and Abort Detection Time Probability Functions........ 33
5. Representative Example of Mode Logic for Vehicle Case.................... 34
6. Derivation of Transition Probabilities for Vehicle Mode.................. 38

INTRODUCTION

The attrition of large numbers of fixed and rotary wing aircraft during the Southeast Asia conflict has focused attention on the requirement to provide design features that will enhance the survivability of these systems, curtailing the losses in hostile engagements. Implementing the design features, whether in initial design or the more costly retrofit programs, usually impacts aircraft weight and cost. Because of these impacts, trade studies are required to develop a prioritized listing of survival enhancement features that increase mission effectiveness. The priority list will identify those features that provide the greatest increment to improved effectiveness, and provide the data base from which a bound can be established on the amount of hardening that is effective from a mission-cost standpoint. The data base will also permit a determination of the best mix of hardening features when a budget constraint is imposed.

The MISDEM (Mission/Damage Effectiveness Model) is a simulation of an aircraft (and its subsystems) experiencing a time-series of events. The events may include targets to be attacked, threat weapons to be encountered, refueling, recovery at an air base, or certain events selected by the user.

MISDEM evaluates the capability and effectiveness of an aircraft system throughout a mission scenario provided by the user. A schematic illustrating where MISDEM fits into the total mission effectiveness analysis procedure is displayed in Figure 1. MISDEM performs a statistical bookkeeping function, aggregating the detailed results of systems and operations analyses carried out for various threat elements and target combinations. The generation of the inputs may require the user to exercise several other simulations. For example, various options of aircraft survivability features would be subjected to trade-off studies using models such as GVAP, GPBP, SHOTGEN, MAGIC, and COVART (JTCG/AS approved programs). An end-game simulation such as ATTACK would then be utilized to produce threat effectiveness numbers to be input to MISDEM. The output of the model is several scalar effectiveness parameters which the user/analyst must assemble and combine to make a determination of mission effectiveness for his particular problem.

The model was developed initially to analyze the impact on system survivability and mission effectiveness of hardening various aircraft subsystems to the weapon effects produced by nuclear weapons. The model, as presented here, has been extended to include nonnuclear weapons effect capability. The approach is an extension of concepts developed by the WSEIAC (Weapon System Effectiveness Industry Advisory Committee).[1] The basic theory of MISDEM has been described by Rockwell International.[2,3,4]

---

[1] Air Force Systems Command. Weapon System Effectiveness Industry Advisory Committee (WSEIAC). Final Report of Task Group II, Andrews AFB, DC. January 1965, (AFSC-TR-65-2 (Volume II)).

[2] Rockwell International (Los Angeles Aircraft Division). *Description of an Improved Effectiveness Model*, November 1973, (TFD-74-62).

[3] Rockwell International (Los Angeles Aircraft Division). *Mission/Damage Effectiveness Model*, 1974, (NA-74-62).

[4] Rockwell International (Los Angeles Aircraft Division). *Mission/Damage Effectiveness Model (Sample Case)*, for Joint Technical Coordinating Group/Aircraft Survivability, Survivability Assessment Subgroup, May 1974, (NA-74-358).

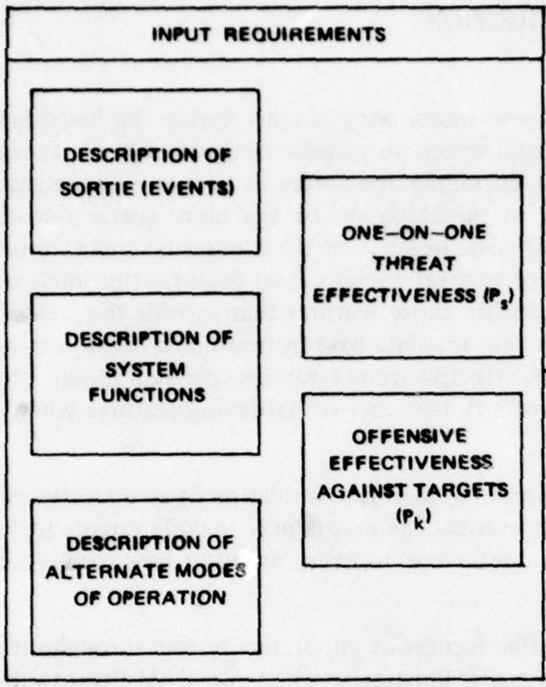


Figure 1. Schematic of Mission Effectiveness Analysis.

Figure 2 is a brief summary of the basic inputs and outputs to the MISDEM simulation. The mission scenario consists of a time/event series, in which the events are either offensive (aircraft system against the enemy) or defensive (enemy defensive system against the aircraft system). The aircraft system must be defined in terms of identifying electronic "black box" functions and mechanical functions that must be performed at specified times during the mission (e.g., terrain following radar, IR target acquisition, engine must operate, and weapon must launch). Each combination of electronic and mechanical functions define system modes of operation. The optimum mode would be to have all functions available and operating. However, due to enemy damage inflicted on the aircraft or system breakdown, some subsystems may not function and the aircraft must operate in a degraded mode. An example of a degraded mode of operation would be the pilot's use of a visual navigation fix plus dead reckoning instead of utilizing inertial navigation with a radar update. Thus, MISDEM is particularly adapted to analyzing aircraft systems that are multiply vulnerable, although singly vulnerable systems are readily handled by the probabilistic nature of the model. In the situation where several functions are lost during the mission, the user may elect to abort the mission and determine the probability that the aircraft can be recovered.[5]

[5] Joint Technical Coordinating Group/Aircraft Survivability. *MISDEM Computer Simulation, Vol. I, User's Manual,* by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles Aircraft Division. Washington, D.C., JTCG/AS, (in process). (JTCG/AS-76-S-003, publication UNCLASSIFIED.)

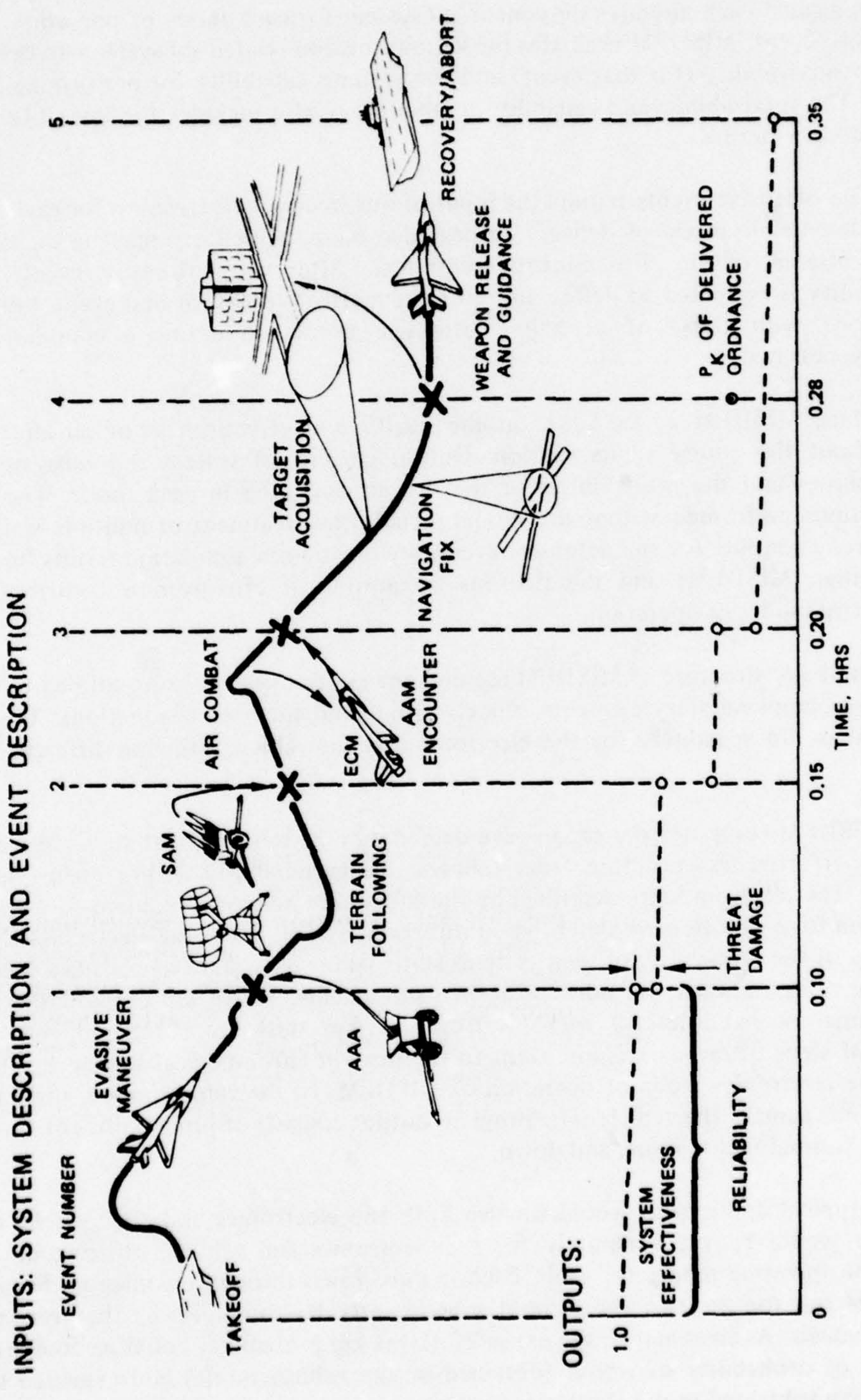


Figure 2. Mission Damage Effectiveness Model Simulation Summary.

Each defensive event requires the input of the effectiveness of the enemy's defensive system against each aircraft component/subsystem for each mode of operation. After each defensive event, MISDEM evaluates the various mission-related subsystems to define aircraft system survivability (for that event) and the systems capability for performing at the next event. The survivability and capability methodology also includes the loss of functions due to reliability factors.

The offensive events require the input of our weapon effectiveness for each target type of each possible mode of delivery (a degraded mode of delivery may be the use of fixed sights instead of the fire control computer). After each offensive event, target kill probability is evaluated to define aircraft system effectiveness for that event; this number is combined with those of preceding offensive events to define a cumulative mission effectiveness update.

Thus, MISDEM *keeps book* on the health and effectiveness of an aircraft system throughout the course of its mission. Output consists of system and subsystem survival probabilities and the probabilities of the system operating in each mode. The important calculations performed within the model include the treatment of multiple warheads (the effectiveness inputs for the defensive events are one-on-one simulation results from a model other than MISDEM) and the rigorous accounting of effectiveness contributions from alternative modes of operation.

The basic structure of MISDEM requires the aircraft system being studied to be divided into two complementary elements: the electronic and the vehicle functions. The MISDEM program is run separately for the electronic and the vehicle, utilizing different inputs for each.

MISDEM computes the progressive degradation in vehicle flight status and electronics mission effectiveness resulting from failures due to reliability and/or enemy air defense actions. The electronics are described by the subsystem networks required for each mode of operation for each offensive/defensive subfunction. The status of the electronics is measured in terms of the probability of each systems state, where each state represents a specific combination of operational and non-operational subsystems. Target kill probabilities and mode operational probabilities for each electronic system state are computed and output as MISDEM steps from one mission event to the next in chronological sequence. This constitutes the electronics mode of operation of MISDEM. In the vehicle mode, there is but one subsystem, namely the vehicle. The program output consists of probabilities of three vehicle modes: normal flight, abort, and down.

A typical application would involve both the electronics and the vehicle; the entire program would be run separately for the electronics and vehicle, utilizing the different MISDEM operating modes for each. Each is thus *flown* through the mission. Results of the MISDEM run for the two cases could subsequently be combined by the user in a higher order analysis. As an example, the expected target kill probability could be computed as the product of probability of arrival (obtained in the vehicle mode) and expected target kill probability (obtained in the electronics mode).

Execution of the MISDEM simulation, which consists of two parts (Program 1 and Program 2), required different interpretation and types of data for the electronics and vehicle modes. The required input data are somewhat different, even though in some cases the same variable names are used.

MATHEMATICAL MODEL

THEORETICAL OVERVIEW

Basically, the function of the model is to propagate the probability distribution of the states of a system. This is done in a recursive (step-wise) fashion. The recursion formula for each state is the Law of Total Probability:

$$P_{j,n} = \sum^{K} (P_{j/i,n}) \times (P_{i,n}) \tag{1}$$

where

$P_{j,n}$ = probability associated with the state of interest at event n

$P_{i,n}$ = probability associated with one of K mutually exclusive states at event n-1 (preceding)

$P_{j/i,n}$ = conditional probability of state j at event n, given a prior system state i

In vector notation:

$$\vec{P}_{j,n} = \begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_K \end{bmatrix} = \begin{bmatrix} P_{1/1} & P_{1/2} & \cdots & P_{1/K} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ & & & \\ P_{K/1} & P_{K/2} & \cdots & P_{K/K} \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \\ P_K \end{bmatrix} = (\vec{P}_{j/i,n}) \, \vec{P}_{j,n-1}$$

where

$\vec{P}_{j,n}$ = vector containing probabilities of all states at the nth event

$(\vec{P}_{j/i,n})$ = transition matrix for the nth event, a matrix of transition probabilities (conditional probabilities)

$\vec{P}_{j,n-1}$ = probability vector at event n-1

Once an initial probability vector is defined, the probability vector at any later event is simply determined by successive multiplications with all transition matrices in the time interval.

The major portions of the computer simulation are for the calculation of the transition probabilities and the generation of the states.

There are two different sets of algorithms in MISDEM for transition probabilities corresponding to two definitions of the system state vector for the electronics mode and vehicle mode, respectively.

The main characteristic that divides system components into each mode is their longevity after being damaged. Electronics mode subsystems include those components that die fast (viz, in the time increment before the next event can happen), while the vehicle mode subsystems are those that die slowly after being damaged (i.e., they may be operable for several time increments before failing). It is possible that some subsystems can be included in both modes depending upon the level of kill to be examined by the user. Thus, fuel cells can be an electronics mode subsystem when catastrophic KK kills are considered and could be included as a subsystem in the vehicle mode when enemy damage causes a leak that results in total loss of fuel several time increments later. The user must define his electronics and vehicle subsystems in a manner that satisfies the situation he is interested in studying.

The common motive in each application is to simulate the loss of significant aircraft functions as a result of exposure to damage possibilities, described in probabilities (discrete or continuous) associated with threat encounters (events) during a mission.

ELECTRONICS MODE

General

In this mode, the program computes an average value of the system payoff variable (measure of system effectiveness) where the average is taken over all system states. The measure of effectiveness is the number of targets killed. The average target kill probability for each target reflects the multiple alternate delivery methods, their individual effectiveness (P_k) and their probabilities of use (related to the system state probabilities). The probabilities of operation with various combinations of subsystems are also outputs of the model.

Outputs of the Program

The expected number of targets killed, given N targets assigned, is:

$$E_T(N) = \sum_{n=1}^{N} P_{k,n} \tag{2}$$

where $P_{k,n}$ is the expected kill probability in a single event. This output parameter is the primary measure of mission effectiveness. The expected kill probability at a given event is:

$$P_{k,n} = \sum_{J=1}^{JCAP} P_{k,J} \times P_{J,n} \quad (3)$$

where

J = weapon delivery mode sequence number

$P_{J,n}$ = probability of the Jth mode sequence in the nth event

$P_{k,J}$ = kill probability in Jth delivery mode sequence

$JCAP$ = number of mode sequences

This output parameter is of interest as an intermediate result. The probability of the Jth mode sequence in the nth event is:

$$P_{J,n} = \sum_{j=1}^{K_{max}} P_{j,J} \quad (4)$$

where

$P_{j,J}$ = probability of state j, which supports mode sequence J

K_{max} = number of states

Modes and Equipment

The behavior of the aircraft system at each event is dictated by the function/ equipment list that is input by the user. The function/equipment list defines the equipment (subsystems) necessary for the execution of each function. It may be possible to carry out a single function with several different combinations of subsystems. For example, there may be three ways to deliver unguided iron bombs onto a target by using different combinations of radar information, barometer readings, on board computer, and iron sights depending on the information supplied by the subsystems that are working. Each combination of subsystems that allows a mission function to be performed is considered a mode of operation (as opposed to the more general electronics mode and vehicle mode applied to ways of using the MISDEM model). Thus, the user specifies for each mission function the possible modes of operation and, where needed, the effectiveness of the system when operating in each mode. It is assumed that the user will identify the best mode for a function as the one with the highest effectiveness and all other modes available for performing that function will be of lesser effectiveness (i.e., degraded modes of operation).

Electronics response to a single event generally consists of a number of functions (e.g., detect threat, employ ECM (electronic countermeasure), etc), and for every function, one of several modes is employed (e.g., jamming, decoy, chaff, etc). The selection of the mode is determined by the system state. Each response to an event consists of a mode sequence (one mode from each function). The ensemble of all possible mode sequences is generated at each event, and a probability is computed for each, based upon system state probabilities as shown in Equation (4).

System States

The system state is defined as a viability vector which is a list of the subsystems that are viable (i.e., capable of being used, if turned on) and those that are not. For example, a simple system might have just two subsystems (pilot, engine); then at zero (takeoff) time both subsystems are working (i.e., viable) and the initial system state would be defined by the viability vector (1,1). The subsystems are identified by their position in the vector according to the ordering (pilot, engine). If the pilot is lost, the system state is defined by the vector (0,1). In all, four vector configurations (system states) are possible for this system: (1,1), (1,0), (0,1), and (0,0). Inherent in the input function/equipment list is a priority which ensures that each viability vector gets associated with the highest priority mode among those whose equipment requirements are satisfied. A probability for every system state is computed based upon subsystem survival probabilities and the mathematical rules for combing them. The probability assigned to a mode sequence is simply the sum of the probabilities of system states associated with that particular mode sequence. Since exactly one mode sequence can occur at each event, the sum of the mode sequence probabilities is unity (except as the user may sacrifice some accuracy for computer running time reduction).

The computation of the system state probabilities constitutes the basic function of the program. They are computed at each event in a recursive fashion (see Figure 3) utilizing the state probabilities from the previous event, and computing the new ones based upon conditional probabilities of transition ("TRANS") related to reliability failure rates and subsystem P_S (probabilities of survival) associated with threat warhead encounters, for a given value of CEP (circular error probable) of guidance.

In the program, the viability vector is represented as a binary number:

$$\underbrace{[0\ 1\ 1\ 0\ 0\ 1\ 1]}_{M}$$

where

"1" denotes a viable subsystem

"0" denotes a subsystem that is not viable

"M" equals number of subsystems

Viability is defined (implicitly in the program logic) as "capable of supporting a system function, if turned on".

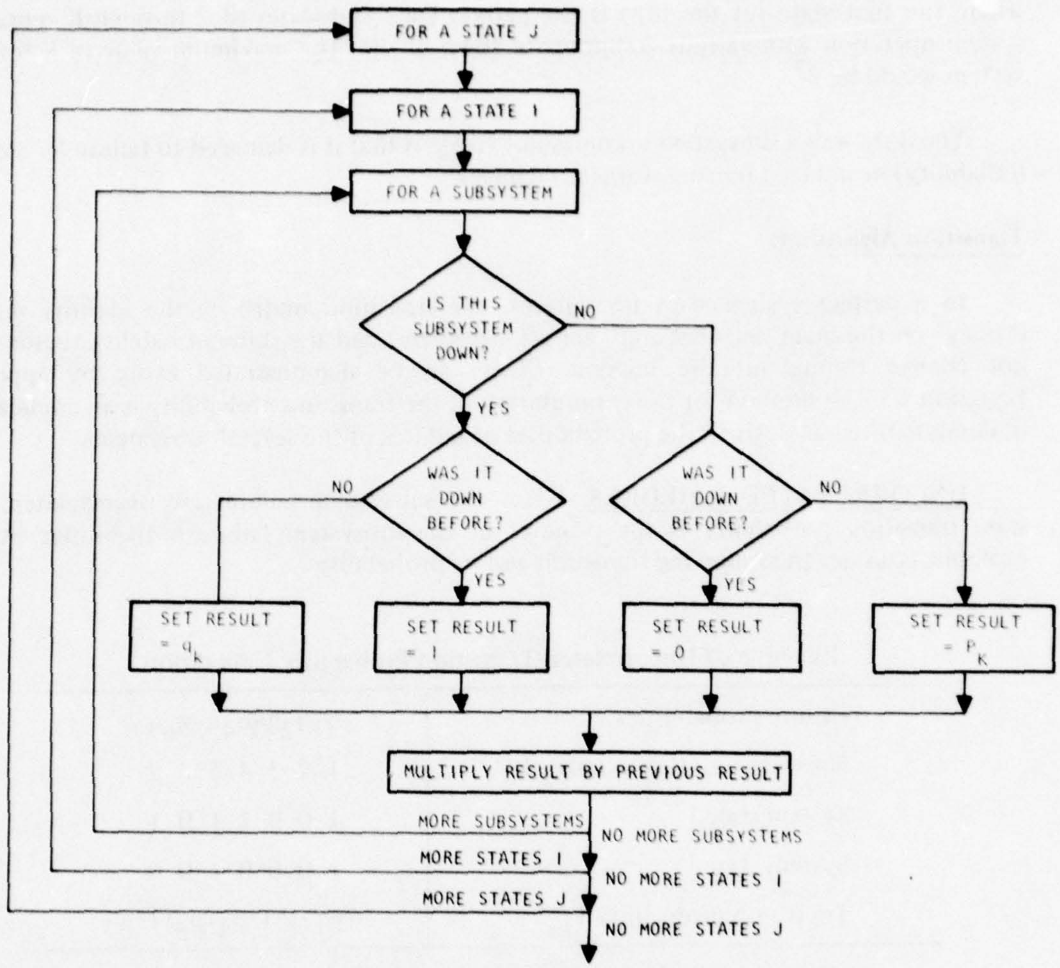


Figure 3. Conventional Damage Transition Algorithm.

The initial state may well be

[1 1 1 1 1 1 1] (the perfect state for a 7 component system)

in which case the initial probability vector is:

$$\begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_K \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

where the first state (at the top) is the perfect state and states of 2 through K represent system operation with various components down or off. The maximum value of K for this system would be 2^7.

The only way a subsystem becomes not viable is that it is damaged to failure by natural (reliability) or induced (enemy warhead) damage.

Transition Algorithms

In a perfect system with no failures, the transition matrix is the identity matrix ("ones" on the main diagonal and "zeros" elsewhere) and the state probability vector does not change throughout the mission. (This can be demonstrated easily by applying Equation 1.) The method for the computation of the transition probability is an application of combinatorial analysis to the probabilities of failures of the several subsystems.

UNCORRELATED FAILURES. When the subsystem failures are uncorrelated, the state transition probability is the product of the subsystem failure probabilities. As an example, consider the following transition and its probability:

Example of Uncorrelated Transition Probability Generation

Failure probabilities	$q_1 q_2 q_3 q_4 q_5 q_6 q_7$
Subsystem ordinal number k	1 2 3 4 5 6 7
System state i	1 0 0 1 1 0 1
System state j	1 0 0 0 1 0 0
Transition probability $P_{j/i}$	$p_1 \cdot 1 \cdot 1 \cdot q_4 \cdot p_5 \cdot 1 \cdot q_7$

where

$$p_k = 1 - q_k$$

(Once a subsystem has failed, the probability of recovery is zero because no repair is assumed during the mission.)

This process is represented by an algorithm that is documented by Air Force Systems Command (footnote 1) (the logic flow is shown in Figure 3).

The transition algorithm for this case has subsystem reliability (in the transition) as a multiplying factor. Thus, P_k is composed of two factors: d_k (reliability) and v_k (survival of threat damage).

RELIABILITY. The factor d_k, representing subsystem reliability in the transition, is obtained from

$$d_k = \exp(-\Delta t \div \text{MTBF}) \doteq 1 - \Delta t/\text{MTBF} + \Delta t^2/2\text{MTBF}^2 \quad (5)$$

where

Δt is the elapsed subsystem time between the last event and the current event

MTBF is the mean time between failures

KILL PROBABILITY. The factor v_k represents subsystem survivability in the transition, which is the same as the subsystem conditional survival of an encounter occurring in that interval starting after $t_{n-1}+$ and ending prior to t_n-. Damage that develops completely during the time interval Δt is referred to as quick damage. This is characteristic of electronic damage. [Damage that does not develop completely until after subsequent damage exposure is termed slow threat damage and is characteristic of vehicle damage (crack propagation, fluid leakage, and fire propagation). This type of damage mechanism is handled by the vehicle mode of MISDEM.]

THREAT WEAPON GUIDANCE ERRORS. The kill computations in the program provide for variations in threat weapon lethality caused by aircraft countermeasures by having the user input the degraded weapon effectiveness.

As shown in Figure 4, the modes in a defensive event are identified with input values of threat CEP. These CEP are used to assign probabilities to the data base of component survival probabilities generated by the one-on-one effectiveness models.

The data base is generated externally to MISDEM by selecting a simulation that can handle a representative array of fragment or projectile sources. For the proximity-fuzed missile warhead, a typical approach for modelling missile-target encounter conditions would be to select an array of offset trajectories (see Figure 5) centered about a representative mean trajectory. Each of these trajectories is then analyzed by an end-game fuzing model to determine the exact burst point location in aircraft coordinates.

Figure 6 shows the offset distribution for a CEP of 100 feet, and the probability associated with a 100- to 200-foot offset that is applied to the calculation of system state transition probability in the "TRANS" block of each event (Figure 3).

The probability of aircraft damage is computed by integrating (over the limits of each zone) the bivariate normal distribution with zero mean. The standard deviation (sigma) is derived from the threat weapon CEP as follows:

$$\sigma = \text{CEP}/1.178 \quad (6)$$

The result of the integration of the bivariate normal distribution over limits R_k and R_{k+1} ($R_{k+1} > R_k$) is:

$$P_{miss}(k) = \exp\,(R_k{}^2/2\,\sigma^2) - \exp\,(R_{k+1}{}^2/2\,\sigma^2) \tag{7}$$

The transition probability associated with such an event is then computed as:

$$P_{j/i} = \sum_{k=1}^{L} P_{j/i}\ (\text{given burst point k}) \times P\ (\text{burst point k}) \tag{8}$$

over all *L burst points described in Figure* 4. One factor in P(burst point 1) is P_{miss} which is applied to all bursts in a given zone. The other factor is the probability of the burst occurring at one of the L points in that zone. This probability is just 1 ÷ L, reflecting an assumed circular symmetric trajectory distribution.

<u>CORRELATED FAILURES.</u> Another major type of damage mechanism that is treated by MISDEM is associated with nuclear weapons (neutron and gamma radiation, blast and thermal effects). In this case, the mutual shielding of one component by another is assumed negligible and the predominant correlation of subsystem failure is: if component "A" is killed, all subsystems with lesser nuclear hardness are also killed. The transition probability algorithm makes use of a lethal radius as a measure of vulnerability of each subsystem, and uses the probability distribution of weapon miss distance normally distributed with zero mean to deduce the probability of a given set of subsystem failures. Figure 7 shows the miss distance density function in r (radius) (the Rayleigh marginal density). The subsystem lethal radii are indicated on the abscissa: the subsystem ordinal numbers have been assigned in decreasing order of lethal radius. The probability of kill of exactly the first k subsystems is the probability that the miss distance exceeds r_{k+1} but does not exceed r_k (equal to the integral of the miss distance density function with lower limit r_{k+1} and upper limit r_k, designated q'_k). An implicit assumption here is that the warhead damage effect is a monotonic decreasing function of distance from the warhead. The following is an example of the computation of $P_{j/i}$ in a case where the least hardened subsystem fails.

Failure probabilities:

1. Natural (reliability)	$q_7 q_6 q_5 q_4 q_3 q_2 q_1$
2. Unnatural (threat)	$q'_7 q'_6 q'_5 q'_4 q'_3 q'_2 q'_1 q'_0$
Subsystem ordinal number k =	7 6 5 4 3 2 1
System state i	1 0 0 1 1 1 1
System state j	1 0 0 0 1 1 0
Transition probability $P_{j/i}$ =	$p'_7 1 \cdot 1 \cdot q'_4 p'_3 p'_2 (q_1 q'_0 + q'_1 \cdot 1)$

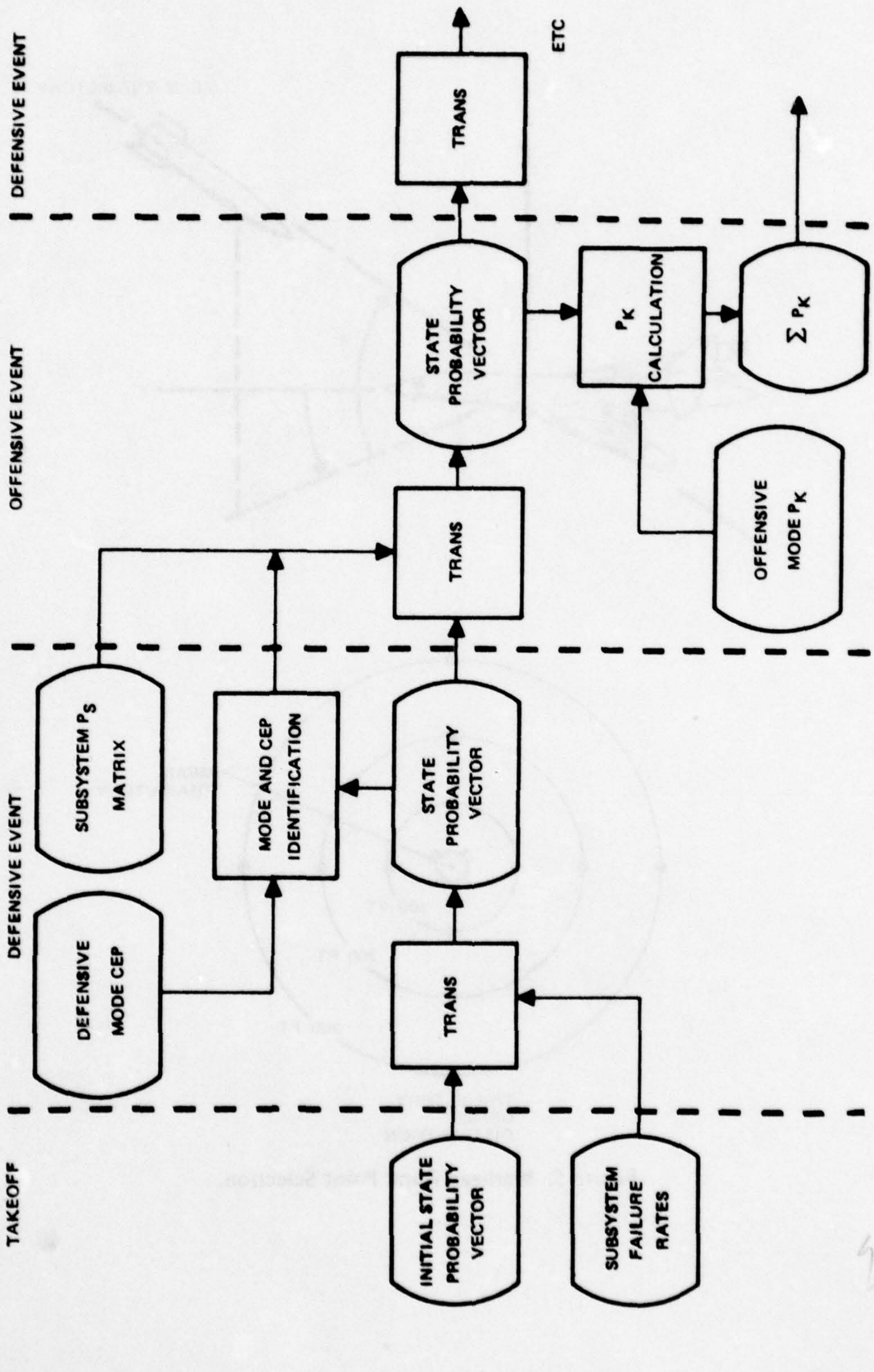


Figure 4. Electronics Mode Data Flow (Example).

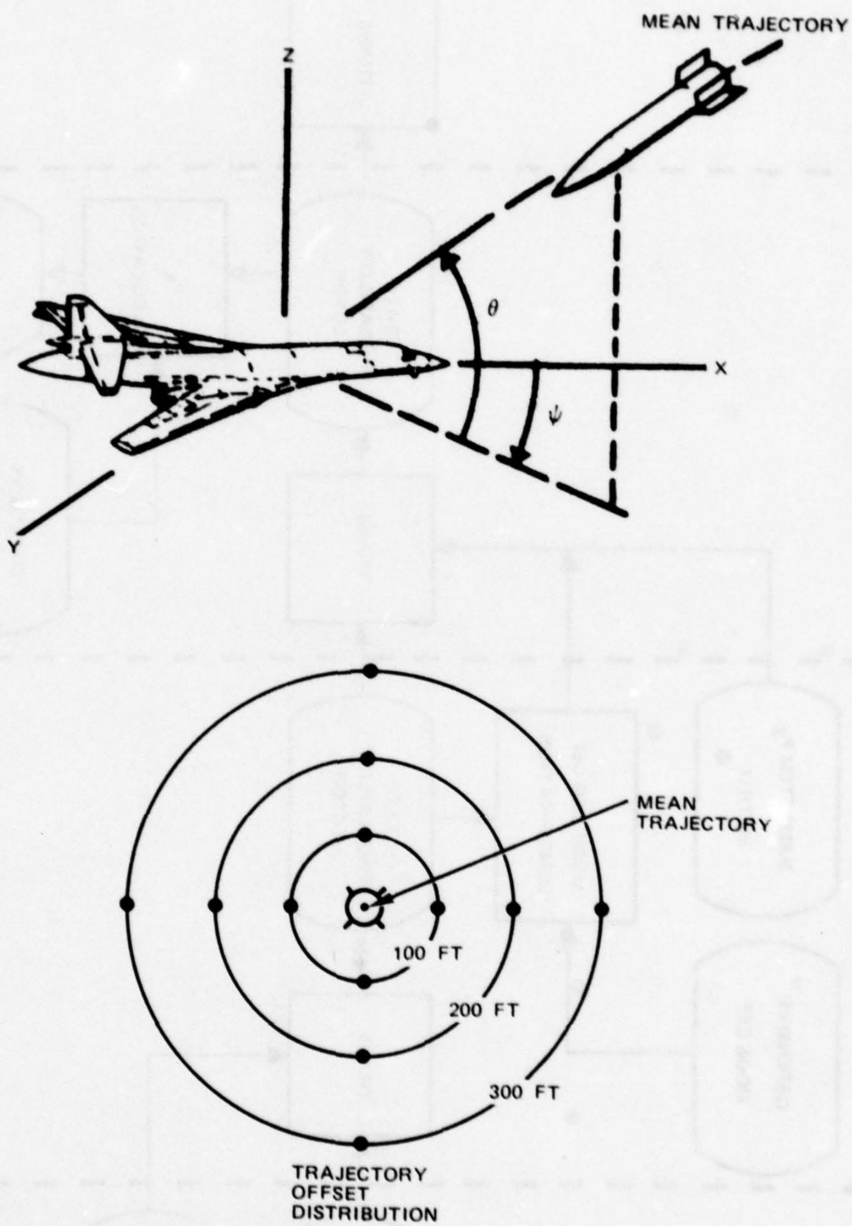


Figure 5. Warhead Burst Point Selection.

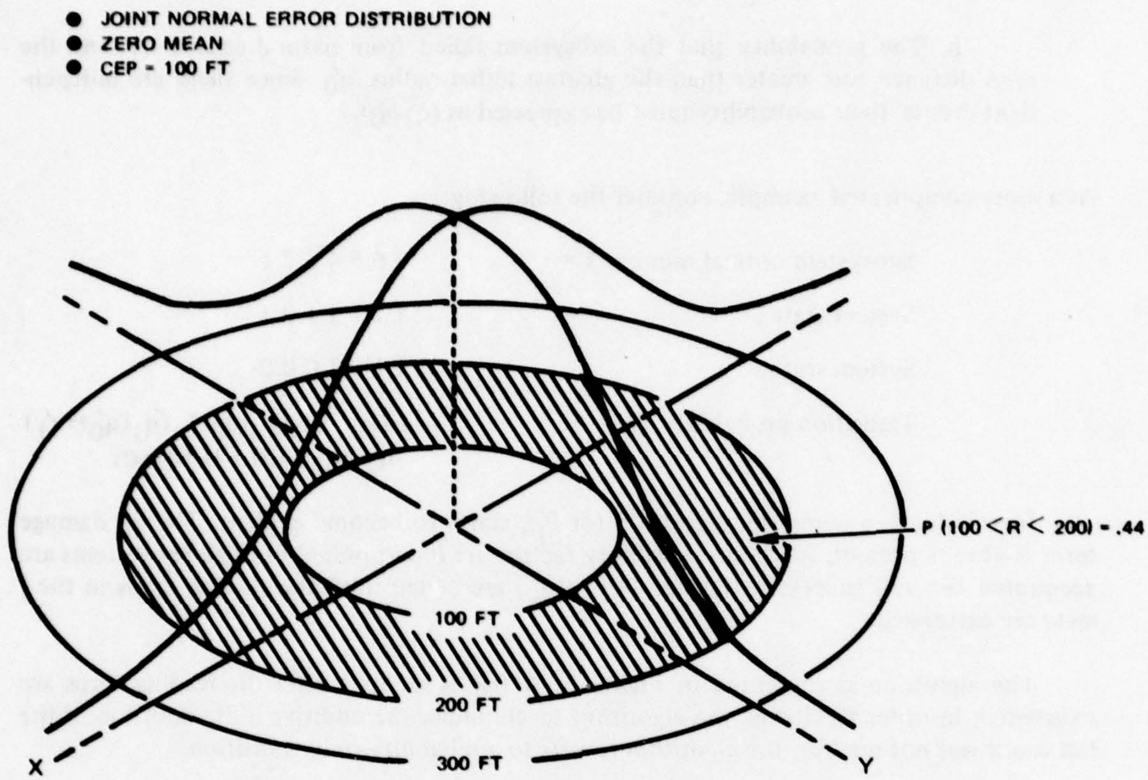


Figure 6. Offset Zone Probability (Example).

The rationale for the final step is as follows:

1. Since subsystem 2 survived the unnatural (threat) hazard, the probabilities of all subsystems of greater hardness (3 through 7) surviving are determined by reliability only. Therefore, in the example $p_7 = 1 - q_7$, subsystems 6 and 5 are not viable and are assigned a transition probability of 1. Subsystem 4 has failed naturally; consequently, the transition probability is noted as one of failure. Subsystems 3 and 2 are treated the same as subsystem 7.

2. Subsystem 1 (the least hard) has a failure probability composed of the sum of the probabilities of two mutually exclusive events.

a. The probability of threat induced kill, expressed as q'_1 (the probability of the miss distance being within the lethal radii for subsystem 1 but more than the radius for subsystem 2) times the probability of the natural failure or survival of the subsystem, $q_1 + P_1$. This may be expressed as $q' \cdot (q_1 + P_1)$ but since $P_1 = 1 - q_1$, this may be shortened to $q'_1(1)$ or simply q'_1.

b. The probability that the subsystem failed from natural causes, q_1; and the miss distance was greater than the greatest lethal radius, q_0'. Since these are independent events, their probability must be expressed as $(q_1 q_0')$.

As a more complicated example, consider the following:

Subsystem ordinal number k =	7 6 5 4 3 2 1
System state i	1 1 1 1 1 0 1
System state j	1 1 0 1 0 0 0
Transition probability $P_{j/i}$ =	$(p_7(p_6(q_5(p_4(q_3(1\ \ (q_1(q_0')+q_1')$ $+q_2')+q_3')+0)+0)+0)+0)$

In this form, a computer algorithm for $P_{j/i}$ starts to become evident. The q_0' damage term is always present, successive reliability factors are incorporated until all subsystems are accounted for and successive damage terms (q_k') are added until the leading zeros in the j state are exhausted.

The algorithm is presented in Figure 8. A flag is set to 1 after the leading zeros are exhausted, in order to change the algorithm to eliminate the additive q'. In addition, if the last event was not nuclear, the algorithm reverts to a reliability-only transition.

The determination of the subsystem lethal radii is based on two factors, illustrated in Figure 9: (1) weapon damage effect, a characteristic of the warhead and geometry alone, measured in pounds per square inch (for blast), calories per square centimeter (for thermal), rads per second (for gamma), and neutrons per square centimeter (for neutrons); and (2) subsystem damage threshold, a characteristic of the subsystem alone, measured in the same units as weapon damage effect.

The definition of subsystem lethal radius is the maximum value of the lethal radii computed for all damage mechanisms, and differs in general from one subsystem to the next due to differences in subsystem damage thresholds. The algorithm which selects this unique lethal radius for each and every subsystem is diagrammed in Figure 10. The lethal radius is updated on every Mth pass through the last block in the figure, where M is the number of subsystems.

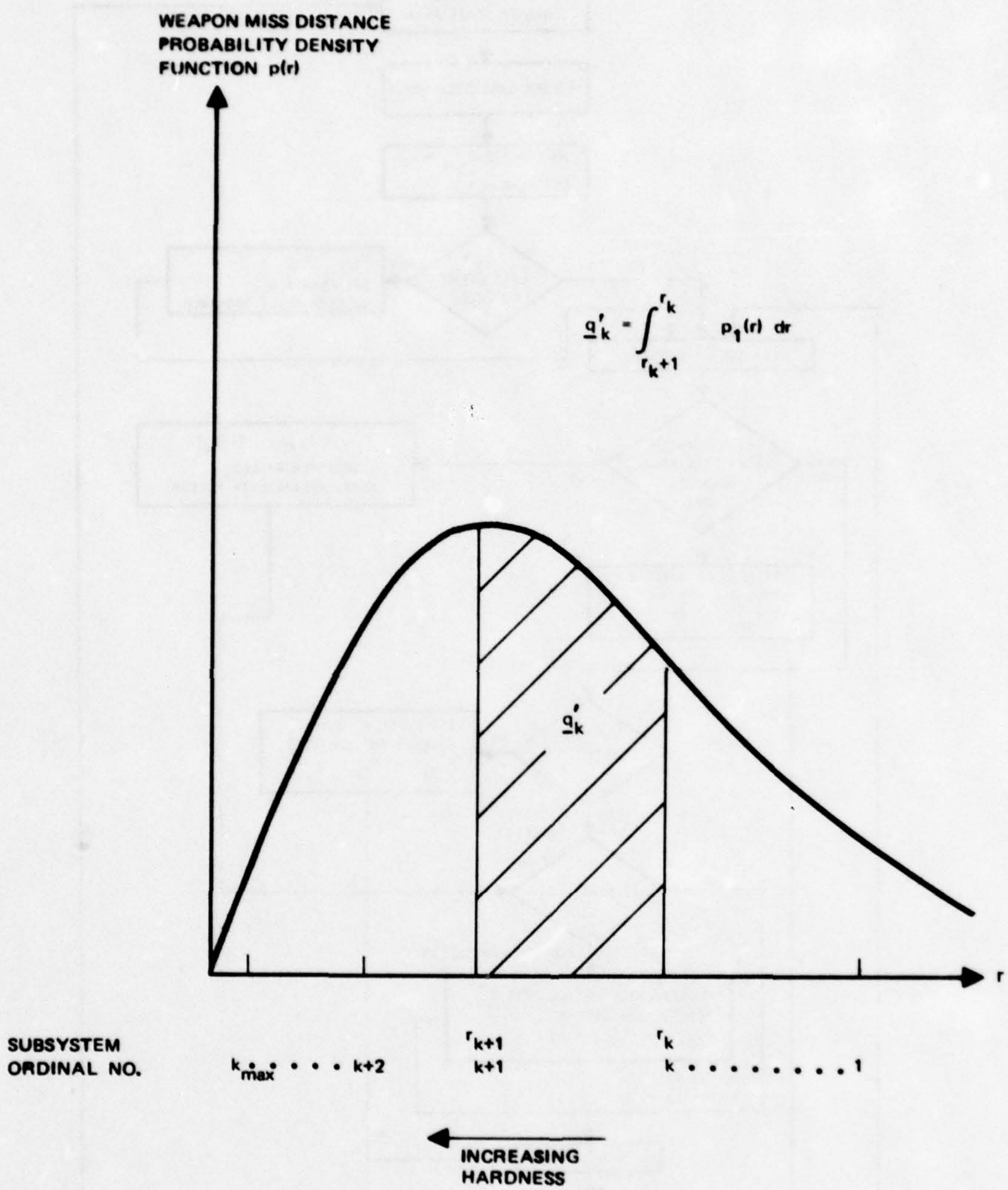


Figure 7. Nuclear Miss Distance Distribution Model.

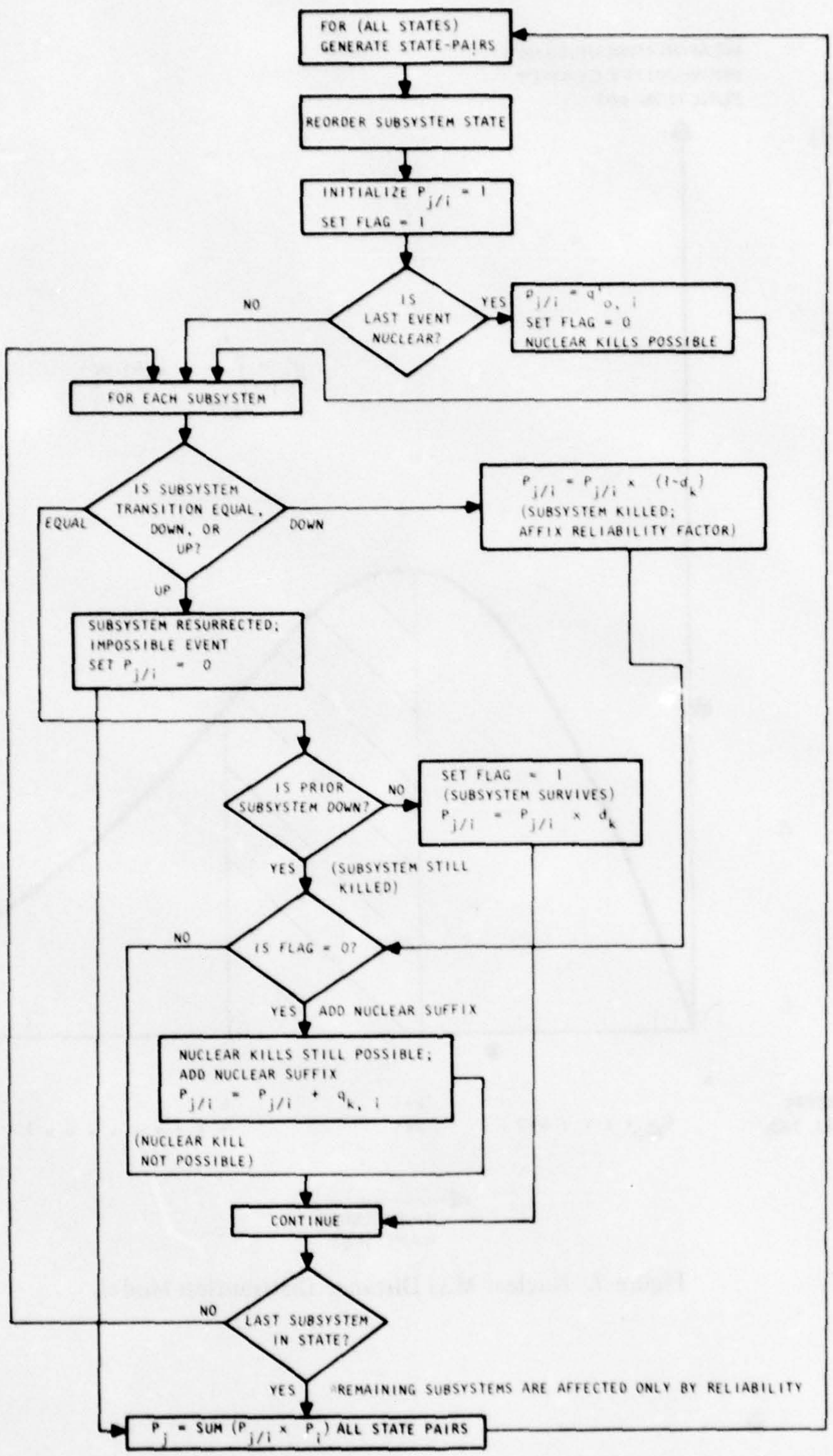


Figure 8. Nuclear Transition Algorithm.

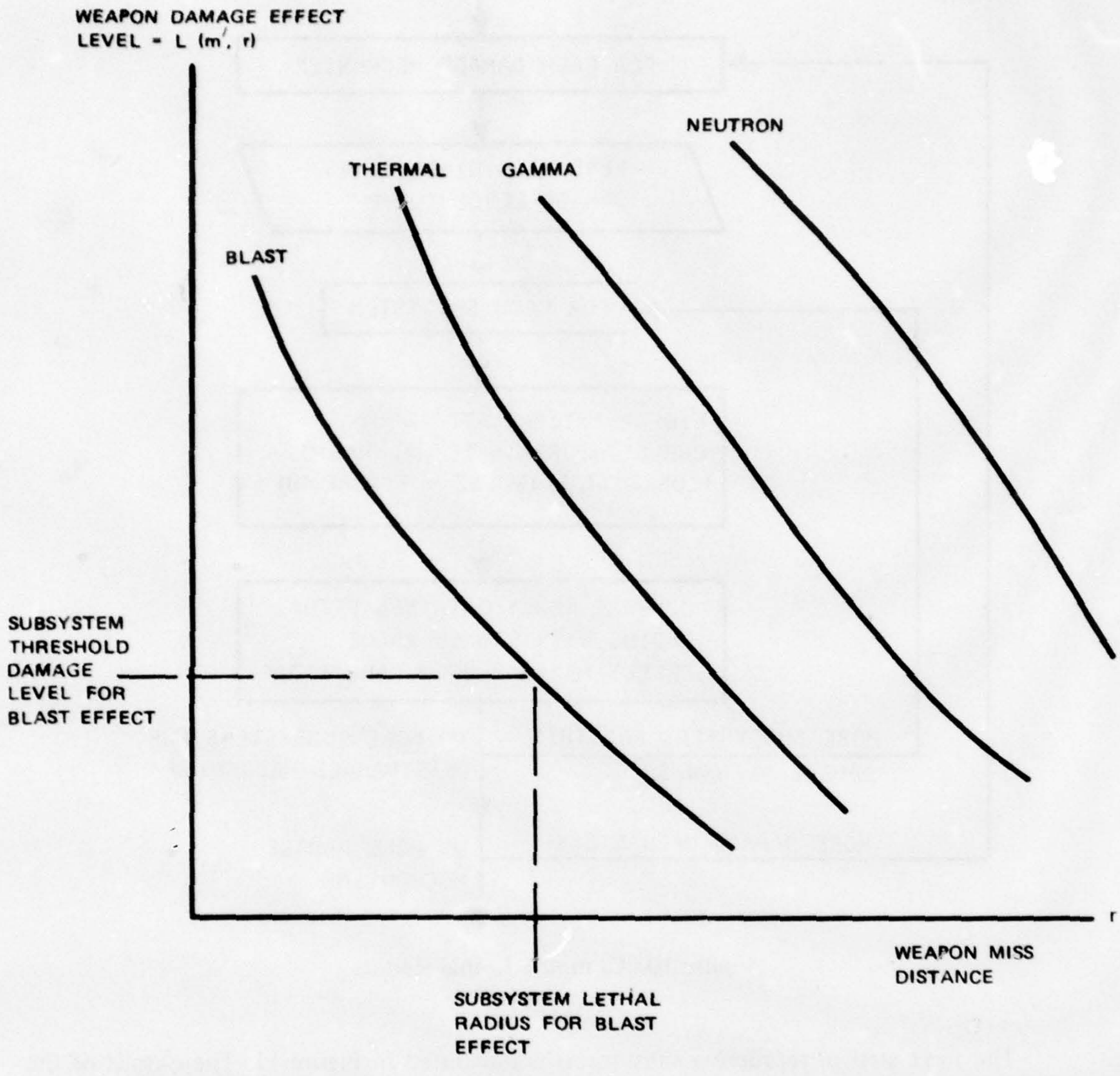


Figure 9. Subsystem – Lethal Radius Model.

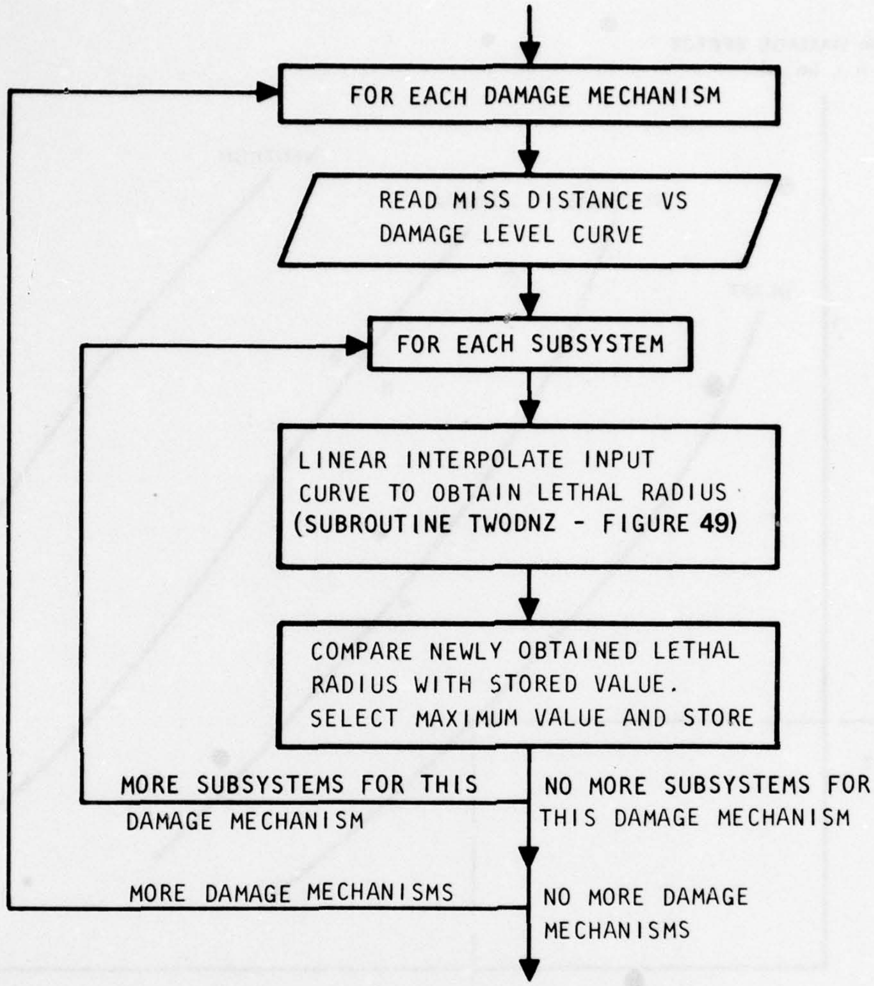


Figure 10. Compute Lethal Radius.

The next step in reordering subsystems is illustrated in Figure 11. The output of the algorithm is a lethal radius RM(k), and an order number IZ(k) for each subsystem number (k) assigned in order of decreasing lethal radius. The program selects the first subsystem, then tests the next subsystem to determine if it has a larger lethal radius. If it does, the lethal radius and the order number of the first subsystem are assigned to argument 2, and the lethal radius and the order number of second subsystem are assigned to argument 1. Therefore, the order number of subsystem 2 is now 1, because it is larger; and the order number of subsystem 1 is now 2. Having selected the largest lethal radius for argument 1, the computer then tests the third, fourth, etc., (to the end) against the first; and if any are larger than the first, additional switches take place. This process of pair-wise comparisons is repeated (using the outer do-loop) to load RM (2) and IZ (2) with the subsystem having the largest lethal radius among those other than RM (1) (which is never tested again), and is

therefore second largest. The process is repeated until each position in RM and IZ arrays are filled. The IZ array contains the vulnerability order number for each original subsystem number. Thus, the subsystem viabilities are reordered by means of the IZ array before the transition probability is computed.

The weapon miss distance density function is a product of probabilities of two random variables ($P_{r/d} \times P_d$):

1. $P_{r/d}(r)$ = conditional miss density (conditioned on the event that the warhead was delivered to the target neighborhood) being a function of the weapon guidance accuracy (including deceptive countermeasure effects) in terms of CEP.

$$P_{r/d}(r) = \frac{r}{\sigma^2} \exp(-r^2/2\sigma^2) \doteq (1.4r/CEP^2)\exp(-.7r^2/CEP^2)$$

The probability that the miss exceeds r_k is:

$$\int_{r_k}^{\infty} P_{r/d}(r)\, dr = \int_{r_k}^{\infty} \frac{r}{\sigma^2} \exp(-r^2/2\sigma^2)\, dr$$

Rearranging the limits to introduce a negative sign under the integral to put the integrand in the standard form of $e^u du$ produces:

$$P_{r/d}(r) = \int_{\infty}^{r_k} -(r/\sigma^2) \exp(-r^2/2\sigma^2)\, dr = \exp(-r_k^2/2\sigma^2)$$

2. Pd = probability of delivery, being a product of probabilities of (assumed) independent events:

a. Probability of weapon launch (a function of detection and launch delays, as affected by denial and confusion countermeasures)

b. Probability of in-flight reliability (guidance, control and fuzing, independent of countermeasures)

c. Probability of no catastrophic guidance denial due to destruction of a portion of the guidance system.

The expected number of rounds delivered is SPd, where S is the maximum number of rounds possible. The expected number is called N.

To reduce the number of program iterations required for simulating a mission, the effects of several (N) weapons encountered in a single threat encounter can be combined into a single effect, assuming the single weapon miss density functions are identical.

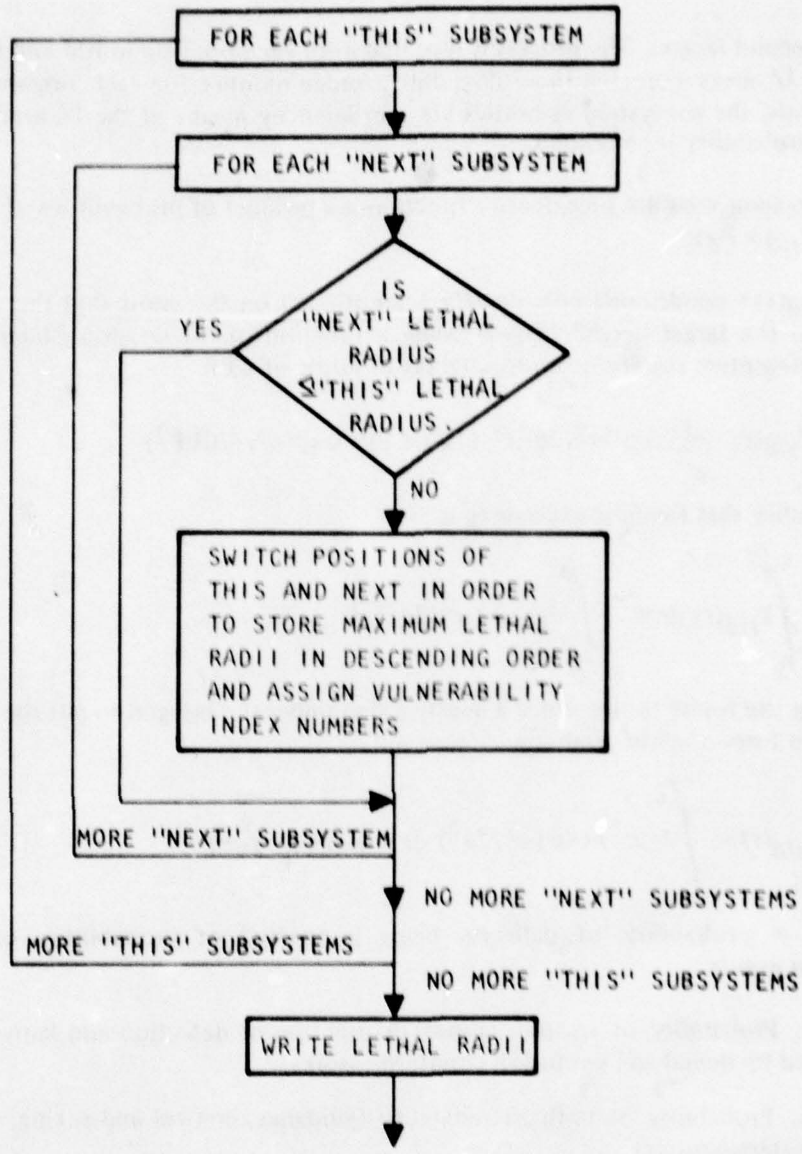


Figure 11. Assign Vulnerability Index Numbers.

The derivation is as follows: N shots are placed in such a way that all N lie outside r_{k+1} (called region AUB) and at least one lies inside r_k (called region A). Under these conditions, (only) subsystems numbered k or less are killed. The probability of this occurring is P (all in AUB and at least one in A)

= P (all in AUB) x P (at least one in A given all in AUB)
= P (all in AUB) x [1 - P (none in A given all in AUB)]
= P (all in AUB) x [1 - P (all in B given all in AUB)]
= P (all in AUB) x [1 - P (all in B and all in AUB)/P (all in AUB)]
= P (all in AUB) - P (all in B and all in AUB)
= P (all in AUB) - P (all in B)

But

$$P\,(\text{all in AUB}) \doteq \prod_{i=1}^{SPd} P\ (\text{ith round in AUB}) = \exp\left(-SPdr_{k+1}^2/2\,\sigma^2\right)$$

and

$$P\,(\text{all in B}) \doteq \prod_{i=1}^{SPd} P\ (\text{ith round in B}) = \exp\left(-SPdr_k^2\,/2\sigma^2\right)$$

Thus

$$q'_{k,i} \doteq \exp\left(-SPdr_{k+1}^2/2\,\sigma_i^{\,2}\right) - \exp\left(-SPdr_k^2\,/2\sigma_i^{\,2}\right) \tag{9}$$

The approximation lies in the use of the expectation of the number of rounds.

The index i denotes the dependence of these parameters on the state of the system from which a transition is to be made. Of course, the significant states of the system in this case are those with different defensive avionics subsystems viable. Figure 12 summarizes the elements of the subsystem nuclear kill probability computation.

State Generation

The total probability associated with the system state is 1.0 (i.e., the system must exist in some state) consequently:

$$\sum_{j=1}^{K} P_{j,n} = 1 \text{ for all n}$$

where

K is the total number of mutually exclusive and exhaustive system states of interest (i.e., whose probability is of interest)

n is the event number

P_j is the probability associated with the state j

Therefore, a requirement for defining states in the model is that they be mutually exclusive and exhaustive.

In the electronics mode, each state is a binary M-vector representing subsystem viabilities. Exclusivity is apparent because each state is a different binary number. To ensure an exhaustive set be identified, it is only necessary to generate all possible permutations of ones and zeros in the (ordered) array. This is done by starting with all ones, then introducing a zero, moving it from one end to the other in the array, then introducing a second leading zero, letting the other run, then a third zero, etc., until M zeros have been used.

For each state generated, two data items are developed in the program: (1) its probability and (2) the mode sequence that the system would exhibit, given that the system had that state. 31

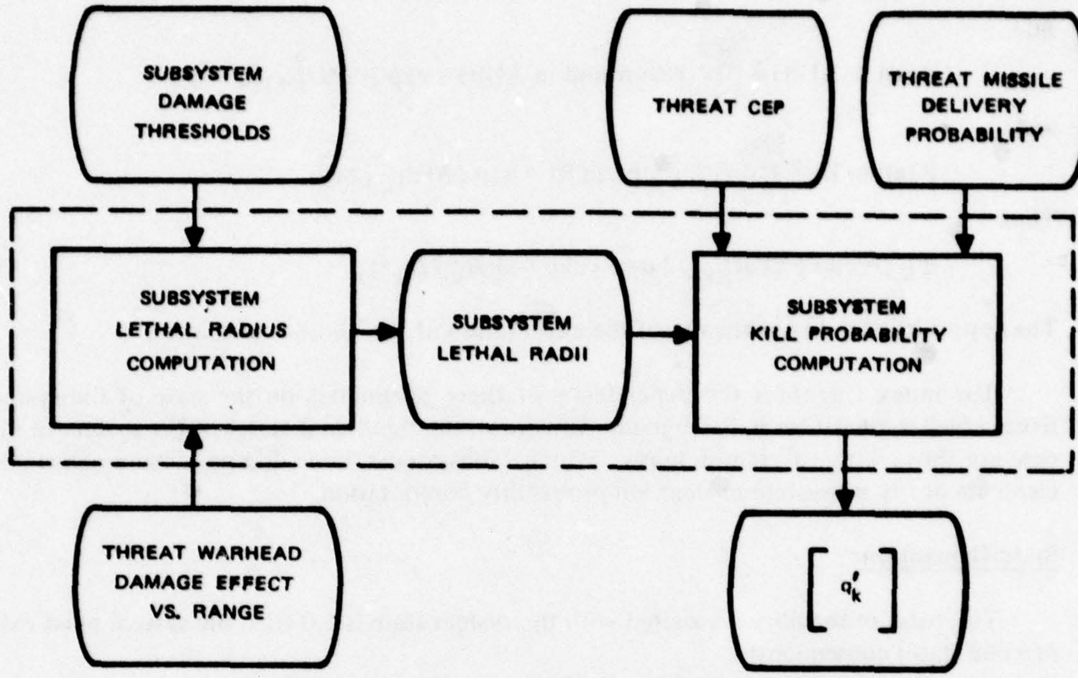


Figure 12. Subsystem Nuclear Kill Model.

Mode Sequence Identification

A sample system configuration is presented in Table 1. The system, as defined for an electronics mode MISDEM run, is composed of thirteen subsystems or components. Some of the components listed may not always be subject to quick killing (viz, engines, pilot, fuel systems and hydraulic components). However, assume for this illustration that we are interested only in assessing the systems susceptibility to KK-kill so that the questionable components do become considered in the electronics mode.

Table 2 is the input which defines, for the four functions to be performed for event 1, the modes and equipments used by the system. Four functions are possible in the example: flight, detection, evasion, and countermeasures. For the flight function, one of three modes must be performed, in order of priority:

1. The normal flight control system must operate (the preferred mode).
2. The backup flight control system must be activated (the second-best mode).
3. The aircraft is disabled.

For the normal flight control system mode, subsystems 1, 2, and 4 through 12 must be viable; however, if this condition is not met, the next mode is examined by the computer, for which only the electromechanical subsystems are required. The first two number columns are ordinal numbers for the indicated function and mode. The second two numbers are employed by the user to specify which of the other functions and modes are to be performed, given the success (i.e., viability) of the first function and mode. The program performs a cataloging operation in which all significant candidate systems states (one/zero configurations) are generated one at a time and each is tested against the mode structure to identify the best mode sequence ascribable to each and every state. In series with the viability test is the mission descriptor test. Each function/mode may require certain environmental or arbitrary program controls placed upon it by the user. This requirement, a T or F (True or False), shows up in the last column. If, as in function 2, mode 2, passive (e.g., infrared) surveillance, a clear air mass is required (indicated by a T in the last column), this function cannot be performed in event 1, since the corresponding value in the event description (F in the last column) indicates that the weather does not cooperate. Consequently, the computer program is directed to another function/mode by means of the last two number columns to the left of the function/mode descriptions.

Table 3 presents the first two mode sequences deduced for event 1 by the computer. The first (best) mode sequence is generated by the perfect system state (all ones in the state vector (not shown)). All distinct flow paths are normally generated. As a check on the user's logic, the corresponding required equipments are also output. The second mode sequence is degraded and results from a loss of subsystems (the jammer being out).

Table 1. Sample System Configuration.

	Equipment	MTBF
1	Left engine	1000.00
2	Right engine	1000.00
3	Sensors/ECM	1000.00
4	Comm/nav/computer/instr/displays	1000.00
5	Pilot	1000.00
6	Fuselage fuel system	1000.00
7	Wing fuel system	1000.00
8	Left master cylinder	1000.00
9	Right master cylinder	1000.00
10	Rudder actuator	1000.00
11	Horizontal stabilizer actuator	1000.00
12	Aft hydraulic power pack	1000.00
13	Electro-mechanical backup flight control system	1000.00

Table 2. Mode Sequence Logic For Event 1.

				Function/mode	Equipment description	Mission descriptor		
1	1	1	2	Flight functions	0000000000000	0	0	F
1	2	2	1	Normal flight control system	1101111111110	0	0	F
1	3	2	1	Back-up flight control system	1101111001101	0	0	F
1	4	5	5	Aircraft down	0000000000000	0	0	F
0	99	0	0		0000000000000	0	0	F
2	1	2	2	Threat detection	0000000000000	0	0	F
2	2	3	1	Passive surveillance	0011100000000	2	4	T
2	3	3	1	Visual mode	0000100000000	2	4	T
2	4	4	1	No detection and no maneuver	0000000000000	0	0	F
0	99	0	0		0000000000000	0	0	F
3	1	3	2	Evasive maneuver	0000000000000	0	0	F
3	2	4	1	Normal mode	1101111111110	0	0	F
3	3	4	1	Limited g-mode	1101111001101	0	0	F
0	99	0	0		0000000000000	0	0	F
4	1	4	2	Countermeasures	0000000000000	0	0	F
4	2	5	5	Jam mode	0011000000000	0	0	F
4	3	5	5	Jammer out	0000000000000	0	0	F
0	99	0	0		0000000000000	0	0	F
999	0	0	0		0000000000000	0	0	F

NOTE: Event description—event 1 is defensive; event occurred 0.19 hours after takeoff; event description is SAM missile encounter F.

Each mode sequence in Table 3 is tested and, if unique, is assigned a unique number. This is done in order that, later in the program, capabilities may be associated with these sequences by number. The process of identifying and numbering the mode sequences is repeated for each state supplied by the state generator. The process is shown schematically in Figures 13 and 14.

Figure 13 shows how the state is tested against mission and subsystem requirements of each subfunction, starting with the first mode in each case. When satisfaction is found in a mode (for each subsystem), its required subsystems are loaded into a "subsystems used" array, and the current subfunction and mode indices are recorded (for the current state) in a mode sequence array.

Figure 14 shows how each new mode sequence array is tested against its predecessors and, if new, a new mode-sequence number is assigned to the current state. If it is duplicated by a predecessor (i.e., is not unique) the old mode sequence number is ascribed to the current state. In either case, the appropriate mode and subsystem lists are associated with the number as indicated (Table 3).

Table 3. Sample Mode Sequences Identified.

Mode sequence	Subfunctional flow
1	Flight functions
	Normal flight control system
	Threat detection
	No detection and no maneuver
	Countermeasures
	Jam mode
	Subsystems used
	Left engine
	Right engine
	Sensors/ECM
	Comm/nav/computer/instr/displays
	Pilot
	Fuselage fuel system
	Wing fuselage
	Left master cylinder
	Right master cylinder
	Rudder actuator
	Horizontal stabilizer actuator
	Aft hydraulic power pack
2	Flight functions
	Normal flight control system
	Threat detection
	No detection and no maneuver
	Countermeasures
	Jammer out
	Etc.

VEHICLE MODE

Outputs, States, and Mode Logic

In this mode, the program computes the probability of arrival of the vehicle at each of a succession of events (threat encounters, targets, home base, etc.). The probability of arrival is defined as the probability that the aircraft did not go somewhere else (as a result of an abort) nor go down prior to reaching that event.

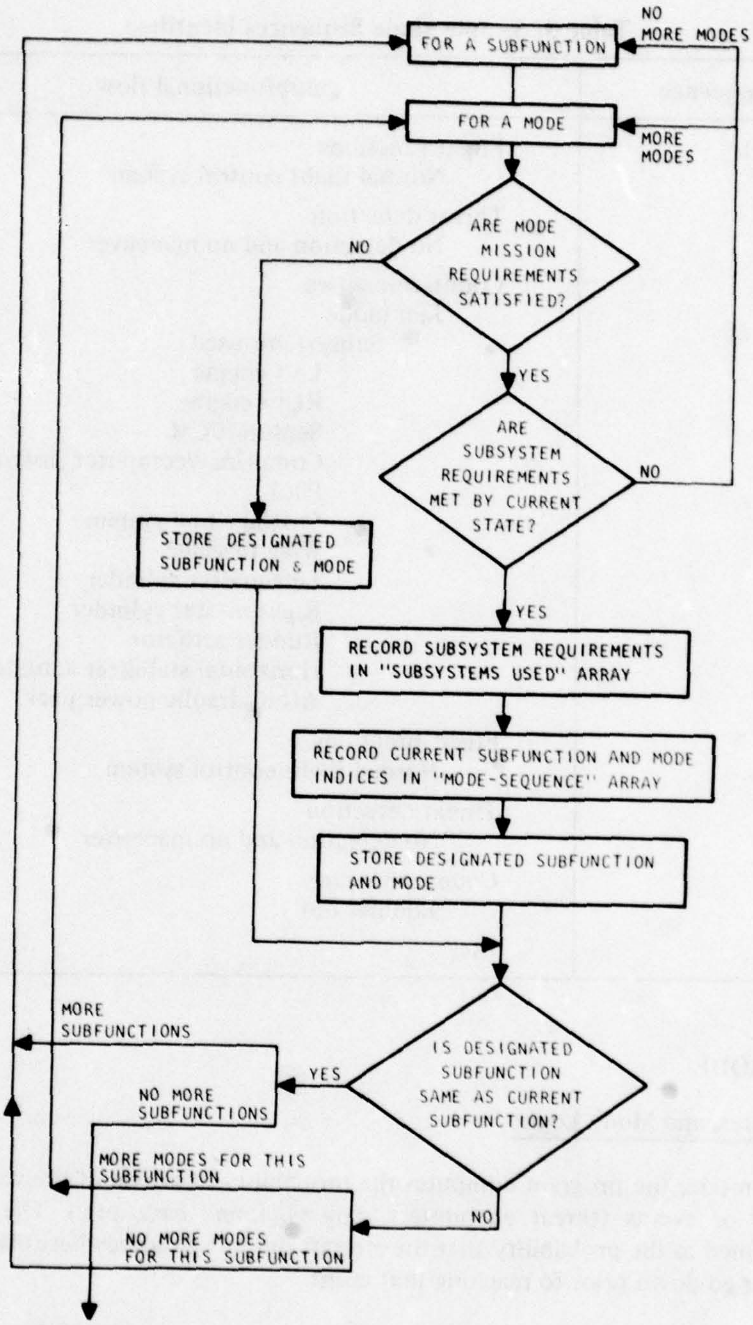


Figure 13. Define Mode Sequence and Subsystems Used.

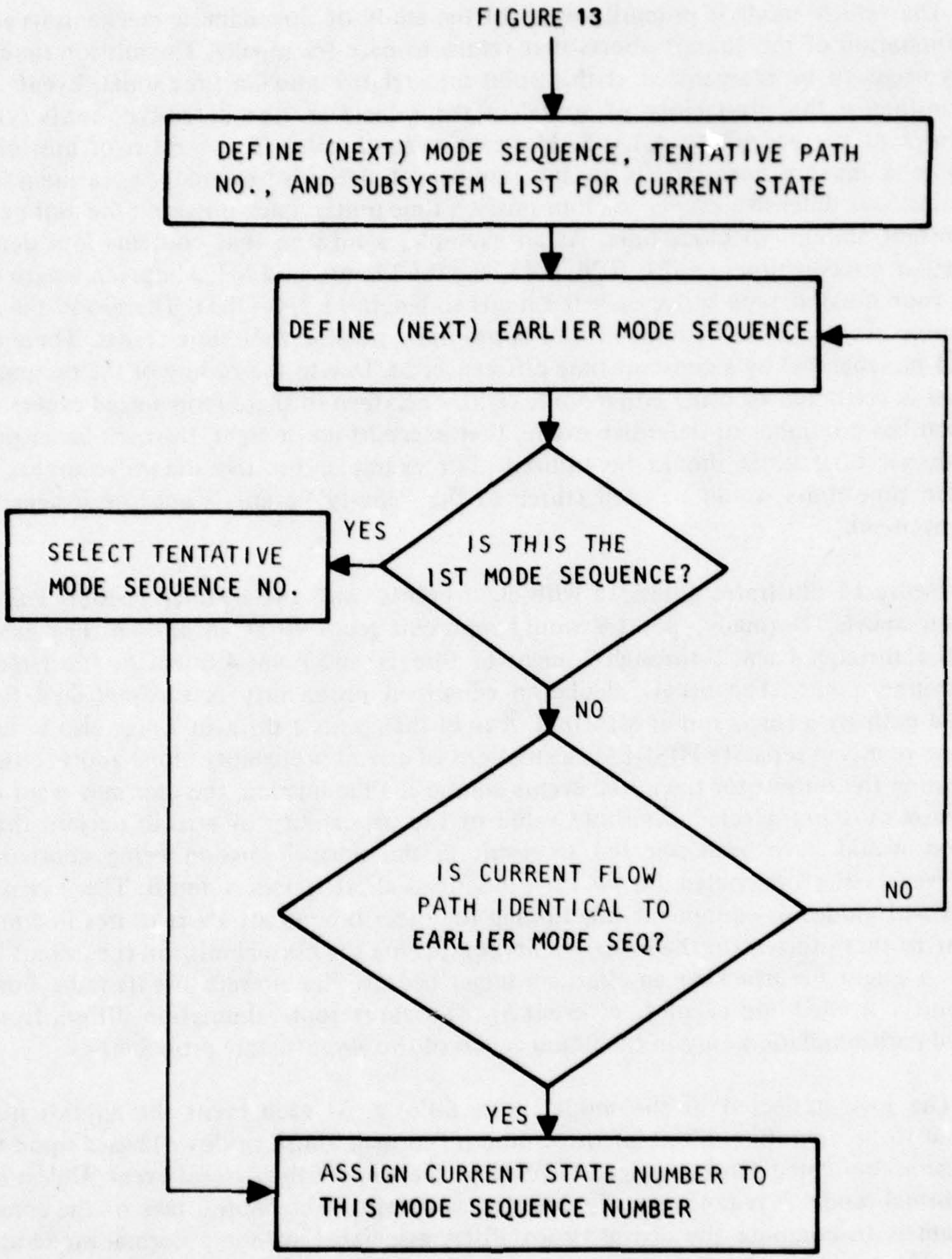


Figure 14. Assign Mode Sequence Numbers.

The vehicle mode is primarily used for the study of slow damage mechanisms and the determination of the aircraft aborts that return to base for repairs. The mission time/event series needs to be reorganized at this point into relative mission time units. Events which will influence the probability of arrival of the vehicle are the defensive events (viz, the exposure of the aircraft to AAA, SAM, or air-to-air threats). The number of mission time units in a mission corresponds to the number of defensive events (e.g., a mission that contains four defensive events has four mission time units). Each mission time unit contains a constant amount of clock-time. As an example, a mission that contains four defensive events, at mission times of 0.1, 0.26, 0.43, and 0.62 hours, and is 1.2 hours in length would have four mission time units each 0.3 hours in length ($1.2/4 = 0.3$). Therefore, the actual defensive events would be lumped into appropriate mission time unit events. These events would be separated by a constant time difference, Δt. Due to the coding of the program, the analyst is restricted to using either four, eight, or sixteen of these reorganized events. If the mission has a number of defensive events that exceed four or eight, the next larger number of mission time units should be utilized. For example, for five defensive events, eight mission time units would be used (three of the "empty" events would be assigned zero effectiveness).

Figure 15 illustrates a mission with eight events, with two of them possibly leading to mission aborts. Normally, point 8 would represent recovery at an airbase. For example, points 1 through 3 and 5 through 7 might be threats, and point 4 might be the target and turnaround point. The usual calculation of arrival probability is accomplished for the normal path by a single run of MISDEM. Any of the points 1 through 7 may also be used as starting points in separate MISDEM calculations of arrival probability along abort routes. By examining the output for the initial events composing the mission, the user may want to, on the basis of a pre-selected threshold value of the probability of arrival, declare that the mission would have been selected to result in the normal mission being aborted. The time/event series designated 1a, 1b, etc., described abort routes A and B. The user defines events and modes of equipment functioning for these time/event abort routes in a manner similar to that utilized for the initial events comprising the normal mission (i.e., event 2a for abort A might be attacking an alternate target because the aircraft lost its radar bombing capability, needed for event 4, at event 3). The abort route simulation differs from the normal path simulation only in the initial values of the aircraft state probabilities.

The logic reflected in the model is as follows. At each event the aircraft may be deemed to be operating in one of three modes (normal, abort, or down) based upon threat damage accumulated (and propagated) prior to the time of the current event. Unless it is in the normal mode, it is not normally counted as arriving. The central task of the computer program is to compute the arrival probabilities associated with the normal mode at each event. The probabilities of abort and down modes are also outputs of the model. The model probabilities are dependent on the probabilities of damage subsequent to exposure. Slow damage mechanisms are assumed to operate, so that the effects of encounters may occur many Δt later. The time convention used is that a defensive event is the time at which the threat exposure is initiated (not concluded), and is the time from which damage effects are referenced.

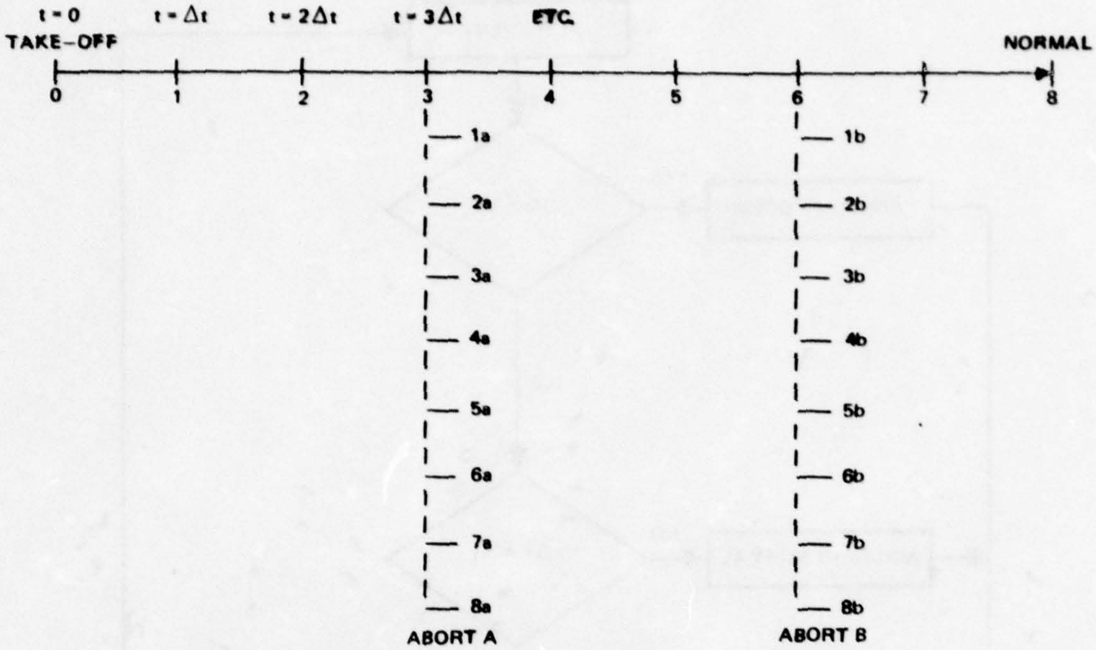


Figure 15. Schematic Scenario for Vehicle Mode, Eight Events.

For each event, there is a cumulative probability distribution of possible times-of-flight (t) remaining relative to the start of the exposure. These times are expressed in mission time units. These probabilities are sometimes known (empirically) for specific times, such as 5 seconds (KK-kill), 30 seconds (K-kill), etc., and may be used to construct the probability distribution (a required input) throughout the mission duration. Depending upon the user (input) mode logic, an aircraft may be classified as down whenever $t < 1$ (t is measured in mission units and is always an integer).

In addition, the aircraft may be classified as aborted when a potentially mission-limiting failure is detected. The time that this detection occurs is labeled τ and is called abort detection time. It is distributed analogously to t and is similarly associated with a threat encounter and measured in mission units. Figure 16 illustrates that logic for classifying aircraft. An example of the two distributions for a single encounter is presented in Table 4. The probability of aircraft failure as a function of mission time units is obtained by cumulating the effects of the defensive events as the mission progresses. The abort probability as a function of mission time units following a defensive event is supplied by the user. Interpretation of the abort probabilities is as follows: considering the third time unit following exposure, the probability of the aircraft failing during this time increment due to accumulated damage is 0.05; and the probability of detecting the need to abort the mission in the same time unit following exposure, is 0.9. However, the damage build up that results in a failure time t = 3 would produce a probability of 0.3 that the crew would detect the need to abort in the first mission time unit following exposure; abort detection probability for time 2 is 0.6, and by time 4 the crew would always detect the need to abort the mission, since the aircraft failed in three time units following exposure.

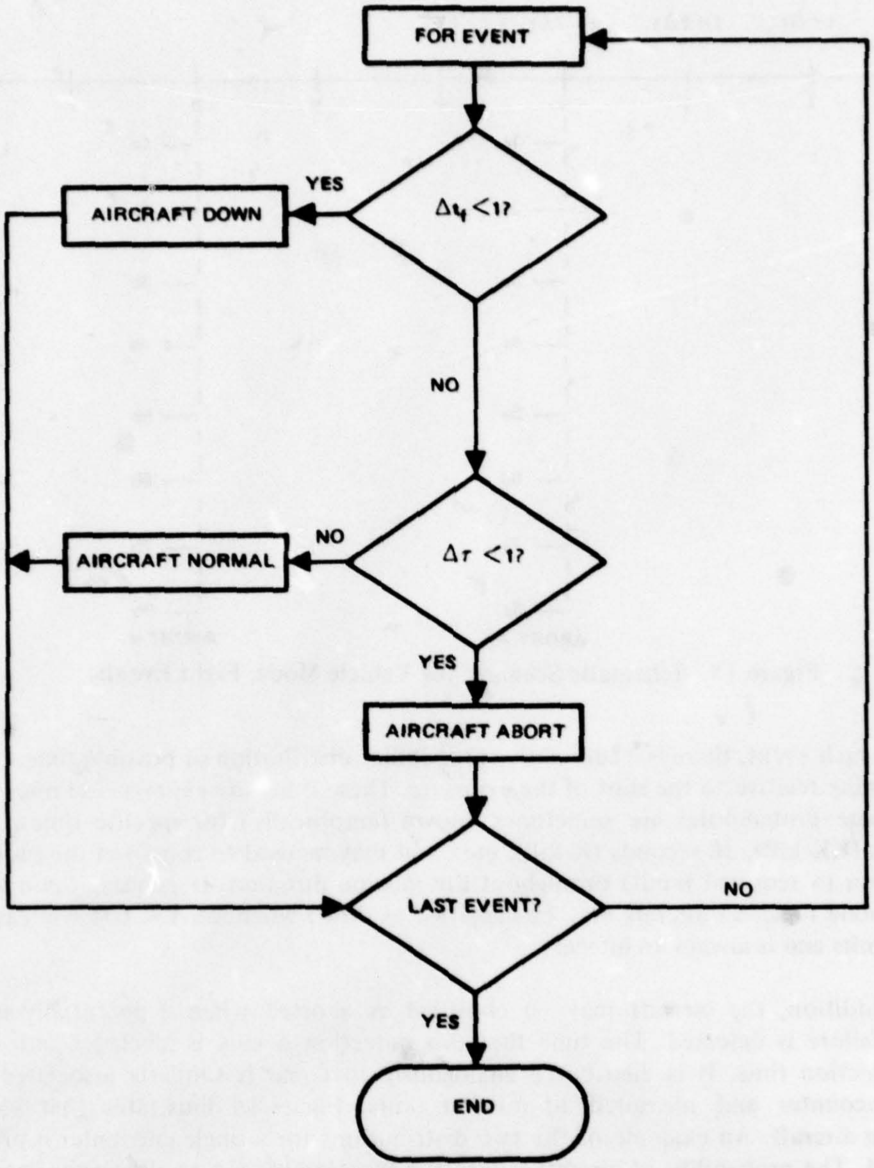


Figure 16. Simple Example of Mode Logic for Vehicle Case.

As shown in Table 4, the abort detection time distribution has been made a function of t, to enable simulating any correlation between damage and its detectability. The flight failure time and abort detection time distribution function are the only means used in the vehicle mode to define the effects of damage. Subsystem failures are simulated only in these terms.

Table 4. Flight Failure Time and Abort Detection Time Probability Functions.

Failure time, t	Probability for t	Abort detection time, τ			
		1	2	3	4
		Probability for τ given t			
1	0.02	1.0	1.0	1.0	1.0
2	0.04	0.5	1.0	1.0	1.0
3	0.05	0.3	0.6	0.9	1.0
4	1.00	0.2	0.4	0.7	1.0

The two times (t and τ) form a doublet that is used as the state variable to compare to failure and abort time requirements. The MISDEM simulation allocates a state probability for each event for all combinations of t and τ. The state probability is then distributed to modes according to the user's mode logic which defines the failure and abort time requirements for each functional mode. A representative example of mode logic is given in Table 5.

Utilizing a mission with four mission time increments, only two binary bits are required to allow description of the four time requirements. The binary numbers and their representative constraints are:

1. 00 represents no constraint on the system.

2. 01 indicates that one time increment must remain in the mission before system failure or the need to abort is detected.

3. 10 represents two time increments must remain.

4. 11 is three remaining time increments.

Therefore, a 01 in the detection time column (τ) of Table 5 means that one time unit remains before the need to abort is detected, and a 11 in the failure time column (t) indicates three units of flight time remain before system failure. The MISDEM programs generate all combinations of the state vectors and compares them for each event with the mode requirements defined by the user. Each state vector that meets or exceeds the mode requirement becomes associated with that mode. Each event must end with all zeros as the requirement (i.e., the most degraded mode) so the probability density properties for each event can be maintained. The probabilities of occurrence for each state are accrued to each mode and event where they satisfy the requirement. Table 5 is further explained as follows:

Event 1. There are no failure or detection time requirements, so no ones appear in the time requirements. The location of ones in this array is analogous to the location of ones in the output format for the electronic sample case, as in Table 1, where "Equipment Description" is replaced by "Time Requirements".

40

Event 2. The first two normal modes require that the time of detection exceeds the previous event duration, so that the vehicle will have arrived at the current event without detecting a need to abort. All time requirements, from the most to the least stringent, must be specified for each mode to keep out probability mass belonging to more-degraded modes. The next mode is also a normal mode, since, even though the damage is detected within the previous event (detection time remaining = zero), the crew determines that the flight time remaining equals the mission time remaining (three units). In the next two modes, the flight time is less than the mission time, so the crew elects to abort. When the flight time remaining after the event is reduced to zero, the aircraft has landed or crashed.

The other events are similar, and moving toward the end of the mission, the criterion for continuing to fly the mission when damage is discovered, is relaxed. When the aircraft, in event 5, has no better abort route than the normal mission route, the abort mode is abolished.

Table 5. Representative Example of Mode Logic for Vehicle Case.

Event	Event description	Mode	Time requirements: Failure time, (binary) t	Time requirements: Detection time, (binary) τ
1	First defensive event	Normal	0 0	0 0
2	Second defensive event	Normal A	0 0	1 0
		Normal B	0 0	0 1
		Normal C	1 1	0 0
		Abort A	1 0	0 0
		Abort B	0 1	0 0
		Down	0 0	0 0
3	Third defensive event	Normal A	0 0	1 0
		Normal B	0 0	0 1
		Normal C	1 0	0 0
		Abort A	0 1	0 0
		Down	0 0	0 0
5	Fourth defensive event	Normal A	0 0	0 1
		Normal B	1 0	0 0
		Normal C	0 1	0 0
		Down	0 0	0 0
6	Landing event	Normal A	0 0	0 1
		Normal B	1 0	0 0
		Normal C	0 1	0 0
		Down	0 0	0 0

Transition Algorithm

Each element of the transition matrix is, as in electronic mode, a conditional probability of transition from a prior state (now a doublet t_i, τ_i) to a current state (t_j, τ_j). The cause of a transition is not necessarily subsystem failure, but rather progressive loss of flight time or abort detection time (relative to the current event time).

An example will show a transition that is not due to decrease of absolute flight time or abort time, and hence is not due to any threat action. Consider the following transition:

	t (binary)		τ (binary)
System state i	0 1		0 1
System state j	0 0		0 0
Transition probability $P_{j/i}$	1	times	1

The transition probability is unity because the prior remaining flight and detection time (due to all prior damage exposures) is reflected in state i, and must be decremented by one mission time unit at the time of state j. There are no other possibilities.

A more complex example will show a transition that reflects a decrease in absolute flight and detection time:

	t (binary)		τ (binary)
System state i	1 0		1 0
System state j	0 0		0 0
Transition probability $P_{j/i}$	$P_f(1)$	times	$P_d(1, 1)$

where

$P_f(1) = P(t = 1) = P(t-1 = 0)$

$P_d(1, 1) = P(\tau = 1/t = 1) = P(\tau - 1 = 0/t - 1 = 0)$

t, τ are the times of flight and detection associated with the preceding event (and measured from that time)

$P_f(1)$ is the probability that t = 1

P_d (1, 1) is the conditional probability that $\tau = 1$ given that t = 1

The transition cannot occur as a result of the time advance alone, since it would result in a transition to the state $(t_j, \tau_j) = (1, 1)$. The only way this transition can take place is that the damage exposure just concluded resulted in a t = 1 and a $\tau = 1$, so that $t - 1 = \tau - 1 = 0$. And the probability $P(t = 1 \text{ and } \tau = 1) = P(t = 1)P(\tau = 1/t = 1)$ by the definition of conditional probability.

The whole transition algorithm is shown in Figure 17. A state-pair is introduced at the top, and tested to reject unwanted states and save computation time. Unwanted states include: (1) states for which $\tau > t$ (considered invalid, since if an aircraft is forced down, the crew is sure to have detected the need by that time), (2) states for which either t_i or $\tau_i \leq 0$, in which case the aircraft was aborted or down in the last event (there is no need to carry them forward), and (3) states for which the current flight or detection time has increased (an impossible transition). All of these state pairs that survive are subjected to one of four transition probability formulae depending on the magnitudes of the decreases in t and τ in the transition. The derivations of these transition probabilities are given in Table 6.

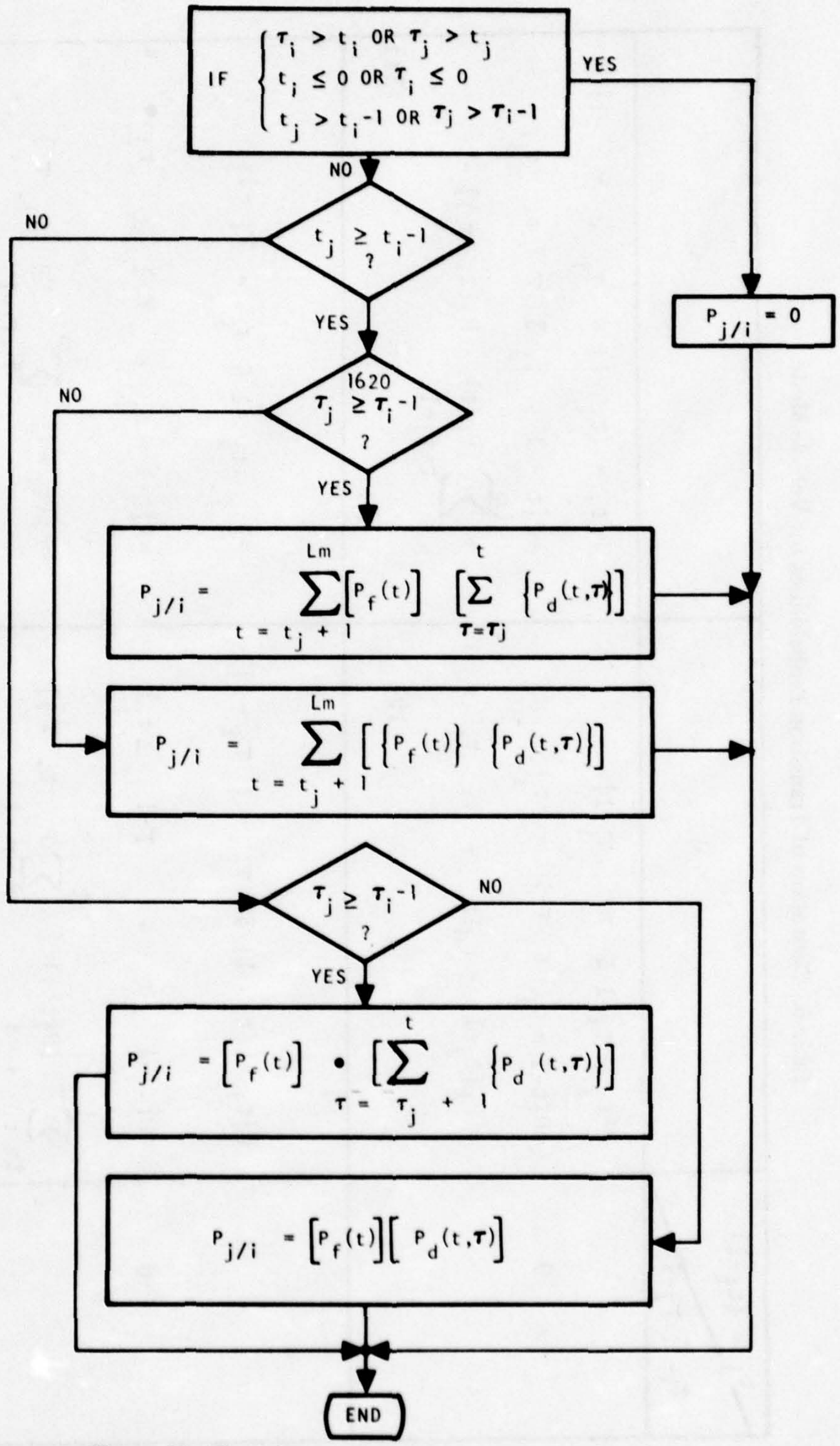


Figure 17. Transition Algorithm for "Vehicle" Mode.

Table 6. Derivation of Transition Probabilities for Vehicle Mode.

$t_j - (t_i-1)$ \ $\tau_j - (\tau_i-1)$	< 0	0
< 0	$P[t_j < t_i-1 \,\&\, \tau_j < \tau_i-1]$ $=P[t-1 = t_j \,\&\, \tau-1 = \tau_j]$ $= P_f(t_j+1) \cdot P_d(t_j+1, \tau_j+1)$ (10)	$P[t_j = (t_i-1) \,\&\, \tau_j < \tau_i-1]$ $=P[t-1 \geq t_j \,\&\, \tau-1 = \tau_j]$ $= \sum_{t=t_j+1}^{L_m} [P_f(t) \cdot P_d(t, \tau_j)]$ (11)
0	$P[t_j = (t_i-1) \,\&\, \tau_j = (\tau_i-1)$ $=P[t-1 \geq t_j \,\&\, \tau-1 \geq \tau_j]$ $= \sum_{t=t_j+1}^{L_m} [P_f(t)] \left[\sum_{\tau=\tau_j+1}^{t} P_d(t, \tau) \right]$ (12)	$P[t_j < t_i-1 \,\&\, \tau_j = \tau_i-1]$ $=P[t-1 = t_j \,\&\, \tau-1 \geq \tau_j]$ $=P_f(t_j+1) \cdot \sum_{\tau=\tau_j+1}^{L_m} P_d(t_j, \tau)$ (13)

List of Abbreviations and Symbols

(Mathematical Model)

Abbreviation or symbol	Equivalent in simulation model	Definition	Units
CEP	CEP(J)	Circular error probable (for threat weapon)	
d_k	PMM(M)	Reliability of kth subsystem in transition	
Δt	DELTAT	Elapsed subsystem time since last event	hours
E_T (N)	ET	Expected number of targets killed (N assigned)	
IZ(k)	IZ(L)	Vulnerability index associated with subsystem k	
JCAP	JCAP	Number of mode sequences	
K	JCOUNT	Number of states	
k	M	Subsystem ordinal number	
L_m	MLTH	Mission length after first exposure	
L(m′,r)	CURVE(I,J)	Designation for nuclear damage mechanism type	Depends on m′[a]
MTBF	MTBF(M)	Mean time between failure	hours
N	NONE	Number of events	
P_i	PI(IJK)	State probability for prior event	
$P_{j,n}$	PJ(I)	State probability for current event	
$P_{j/i,n}$	TRANS	Transition probability, state i to state j	
$P_{j,J}$	PJ(JCOUNT)	Probability of state j (associated with mode sequence J)	
$P_{J,n}$	PCAP(J)	Probability of Jth mode sequence in nth event	
$P_{k,J}$	PK(J)	Kill probability in Jth delivery mode sequence	
$P_{K,n}$	SUM	Expected kill probability (all mode sequences) at event n	
$P_{miss}(k)$	PMISS(K,IJK)	Probability of threat miss in zone k (conventional)	
q_k	PCKILL(M,L,K)	Complement of P_k (conventional threats)	

List of Abbreviations and Symbols (contd)

(Mathematical Model)

Abbreviation or symbol	Equivalent in simulation model	Definition	Units
q_k	QPRM(M,IJK)	Probability of threat miss in kth zone (nuclear)	
r_k	RM2	Lethal radius for kth subsystem (nuclear)	feet
R_K	R(K)	Inner radius of zone k (conventional)	feet
RM(k)	RM(J)	(Maximum) lethal radius for subsystem k (nuclear events)	feet
SPd	FA(J)	Expected number of threat weapons arriving in the neighborhood of the target essentially simultaneously (nuclear case)	
σ	SIGMA	Standard deviation of miss distribution	feet
t	NONE	Time-of-flight due to a particular threat exposure	
τ	NONE	Abort-detection time due to a particular threat exposure	
t_j	LMAT(L)	Current time of flight remaining	
τ_j	LMAT(L)	Current abort-detection time remaining	
V_k	PCSURV(M,L,K)	kth subsystem survival probability	

[a] m'_i=1: gamma dot (rads/sec)
m'_i=2: neutrons (neutrons/sq cm)
m'_i=3: blast (lbs/sq in)
m'=4: thermal (calories/sq in)

SCHEMATIC FLOW DIAGRAMS

This section describes the program architecture, logic, and arithmetic sequence.

OVERVIEW

As shown in Figure 18, the MISDEM program has two parts called Program 1 and Program 2. The reason for separating the programs is that Program 1 is needed to help the user set up Program 2. Program 1 defines all the possible mode sequences that can be obtained from the input mode sequence logic. (This would be very tedious for the user to do manually.) Program 1 also defines a unique mode sequence number (tape output). Program 1 processes one event at a time until event data are exhausted. The user reviews the output and assigns capabilities (offensive and defensive) to each mode sequence number for every event. Program 2 first computes state probabilities reflecting survivability parameters from the last event. These are selectively summed (using the tape input) to obtain mode sequence probabilities. These are then used to compute expected target kill probability (all modes) or system survivability parameters (depending on event type) based on the input capabilities for the current event.

Program 2 also supplies the state probability distribution at each event, which may be used to initialize the state probability for an abort flight path (in a separate application of MISDEM).

In a typical application to a mission, Programs 1 and 2 would be run first in the electronics mode, and second in the vehicle mode. Probability of target kill from the first run would be multiplied by arrival probability from the second run to obtain the expected target kill probability.

PROGRAM ORGANIZATION

Figures 19 and 20 show the principal computation and logic steps in the program in somewhat greater detail than Figure 18, and list all input and output parameters, but leave out all initialization and tape control steps in the interest of simplicity and clarity.

DETAILED FLOW DIAGRAMS

Figures 21 through 49 are detailed schematic diagrams of the simulation program. Figures 21 through 29 describe Program 1, and Figures 30 through 49 describe Program 2. Explanatory text is provided in the "Simulation Model" section for each block of code shown here.

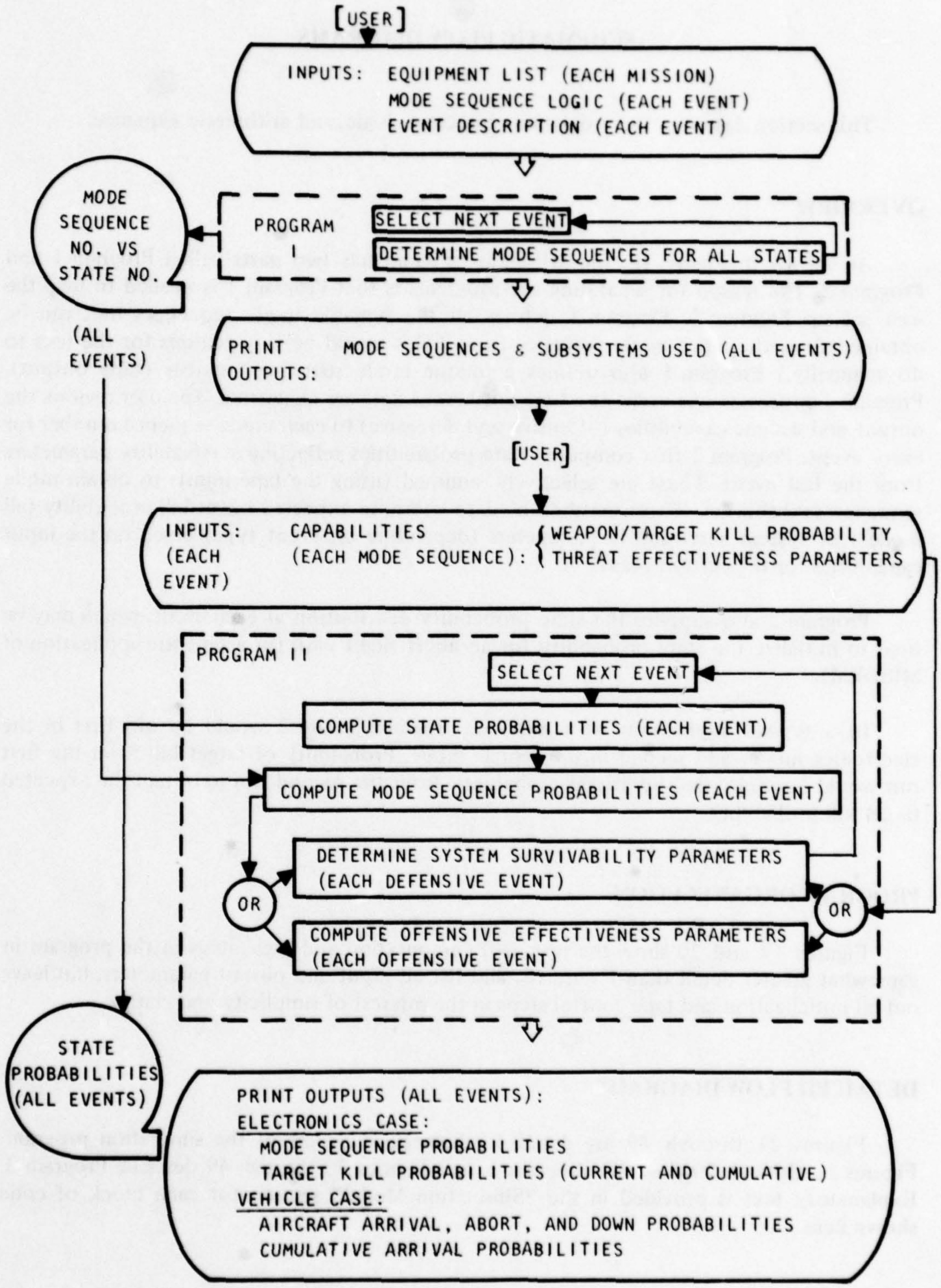


Figure 18. MISDEM Program Simplified Flow Chart.

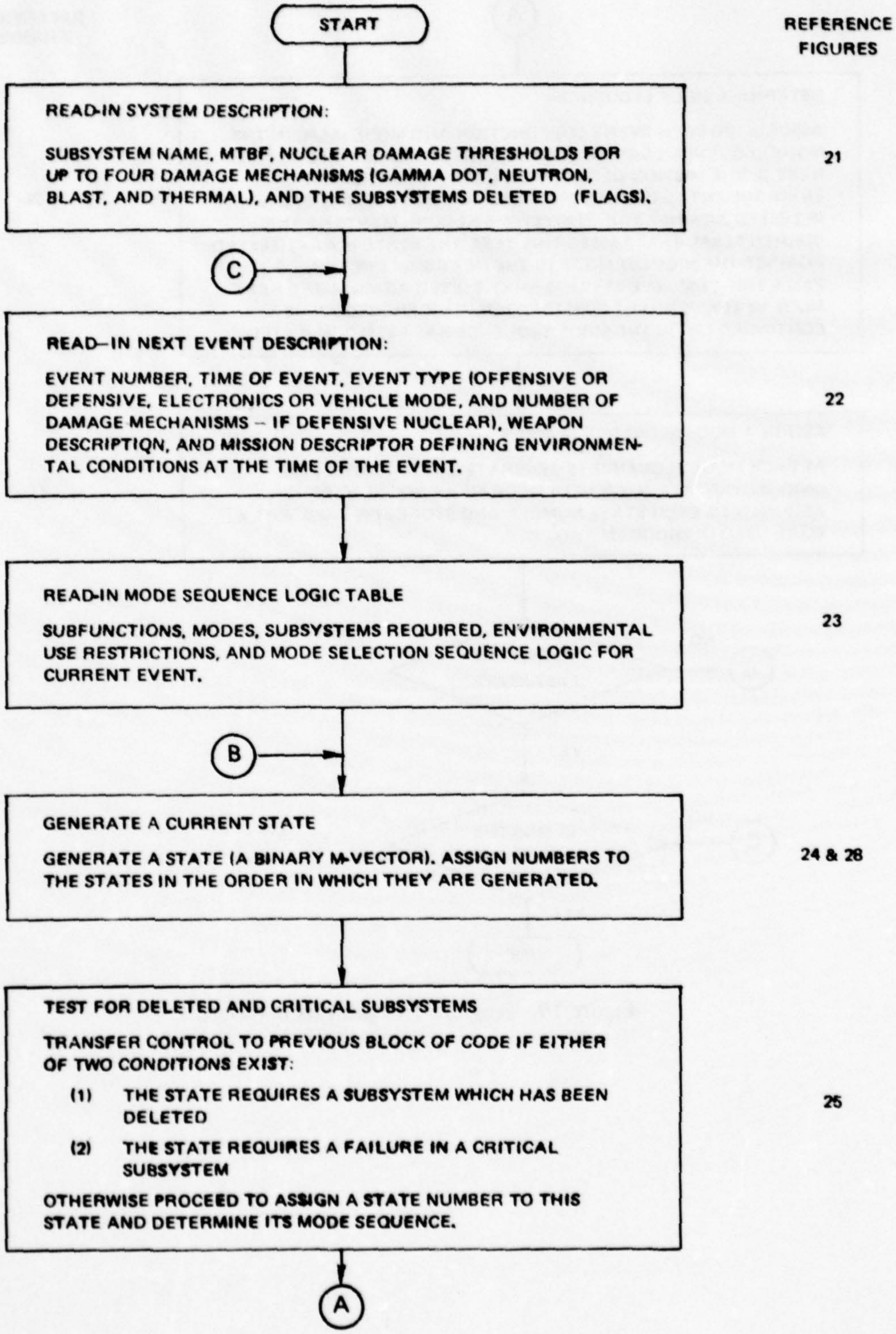


Figure 19. Program 1 Organization.

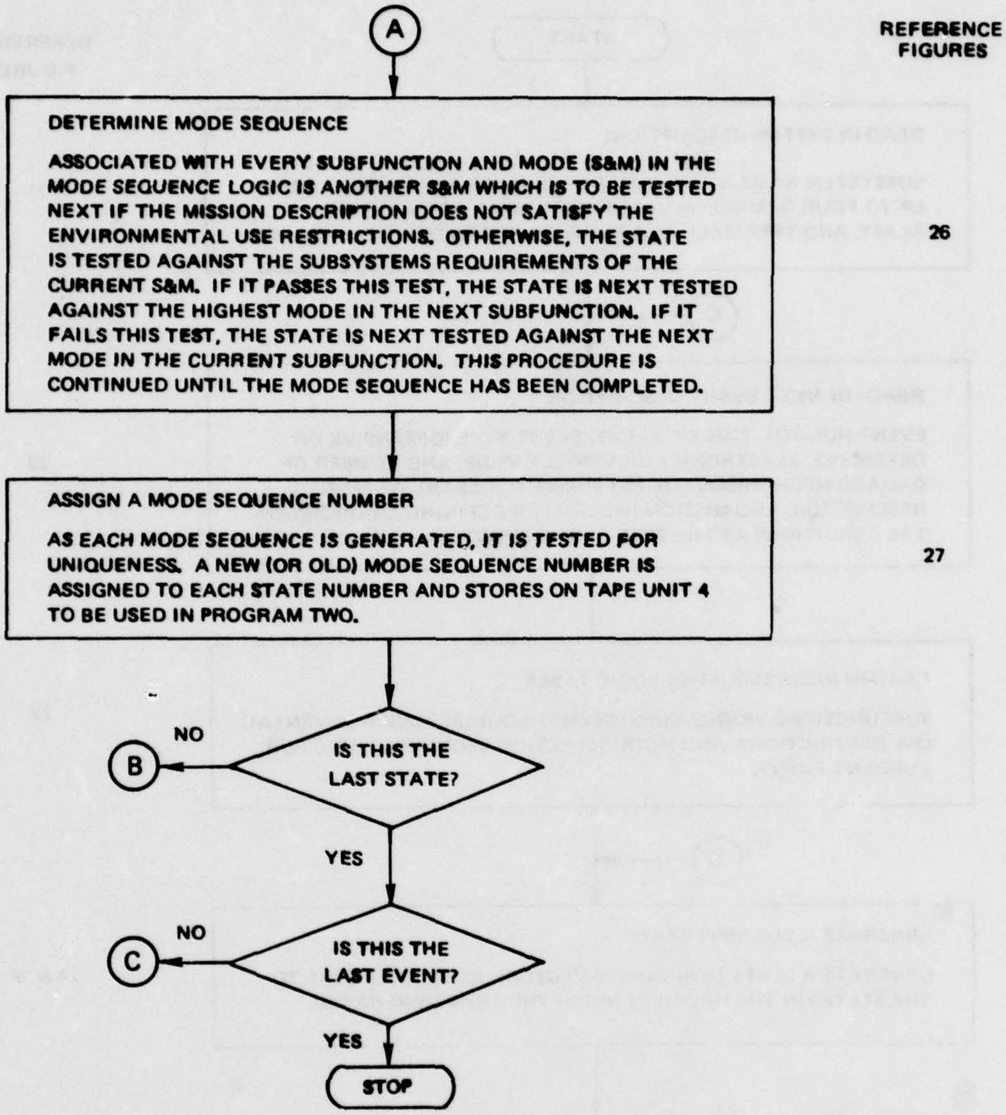


Figure 19. Program 1 Organization (contd.).

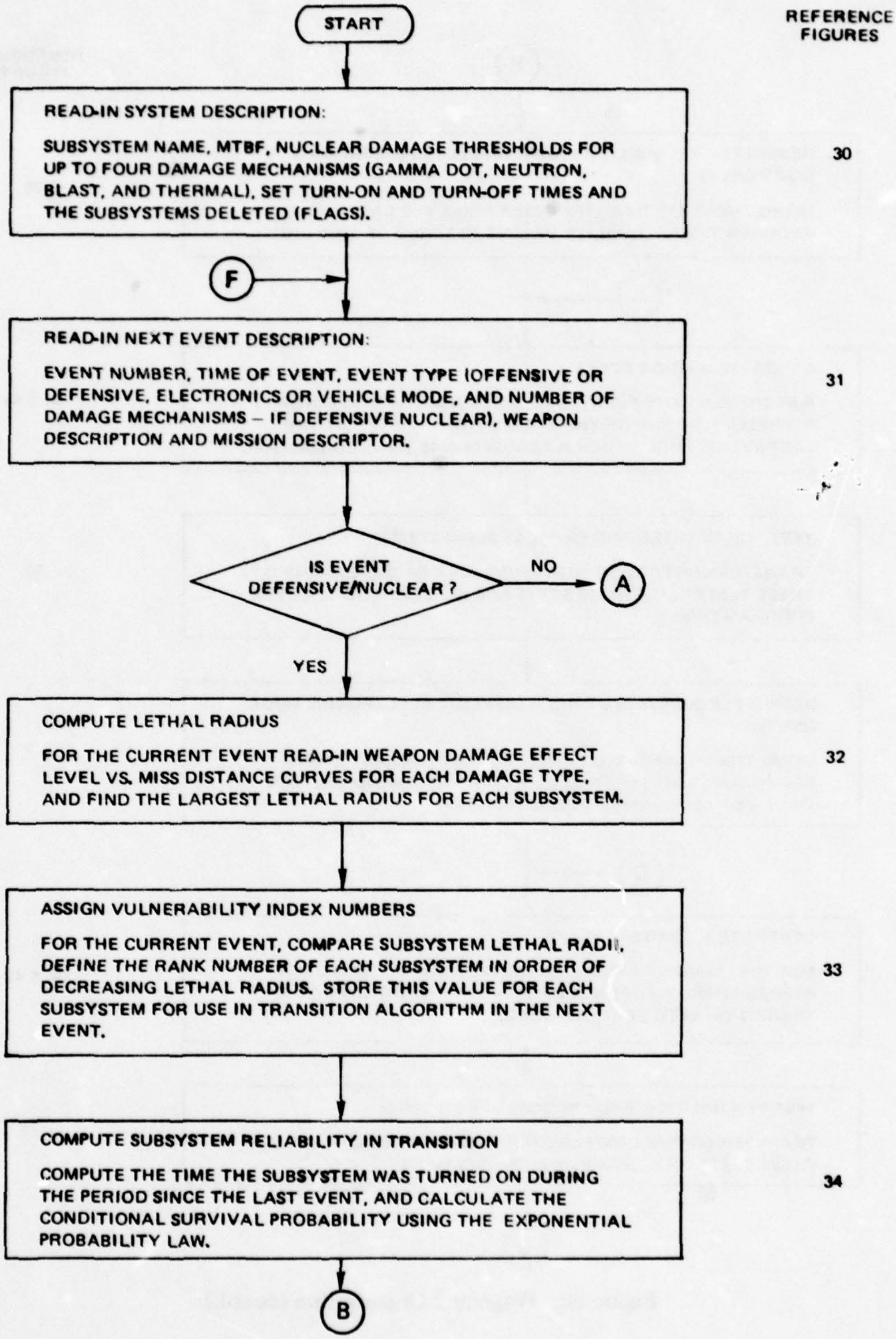


Figure 20. Program 2 Organization.

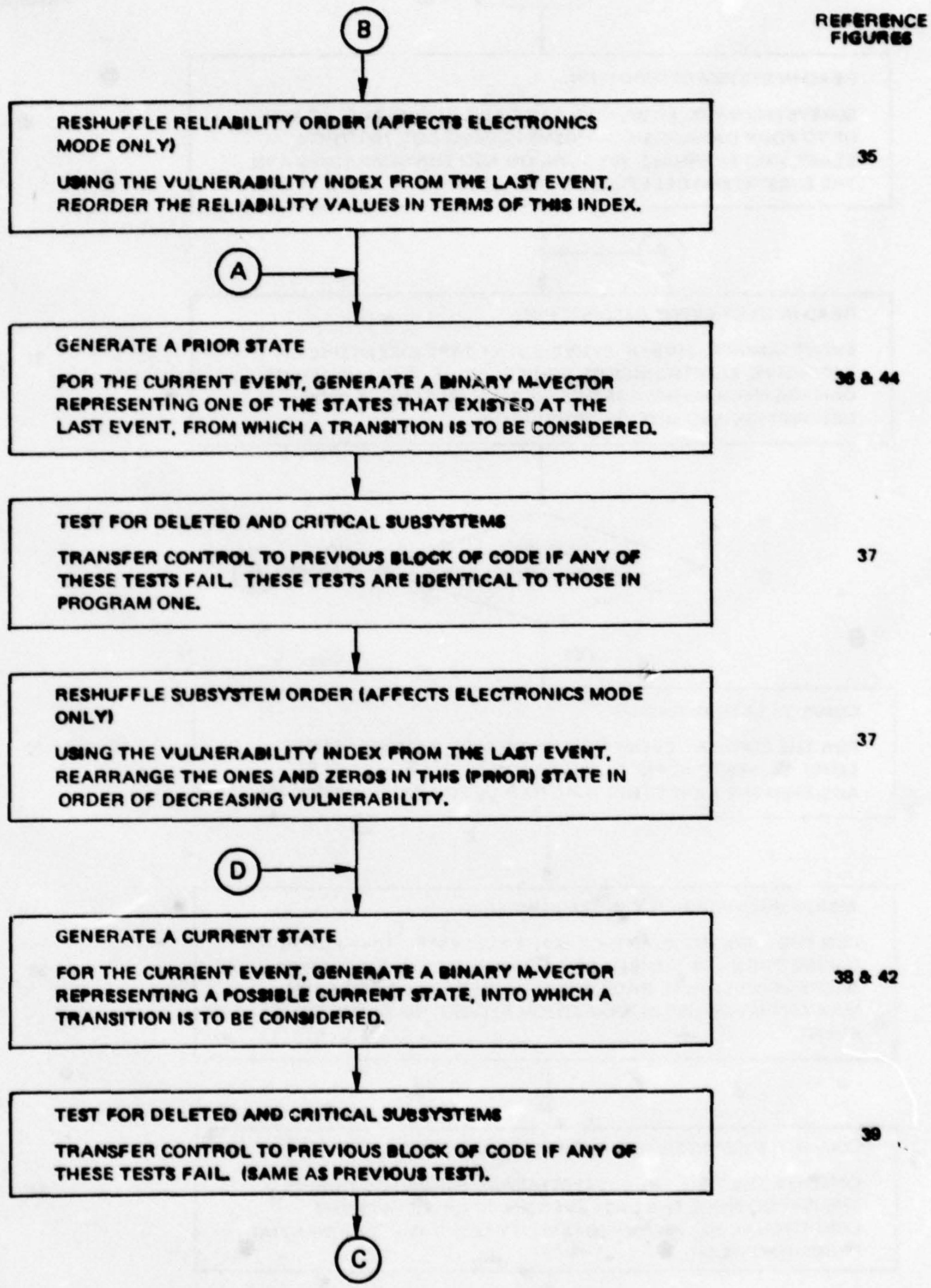


Figure 20. Program 2 Organization (contd.).

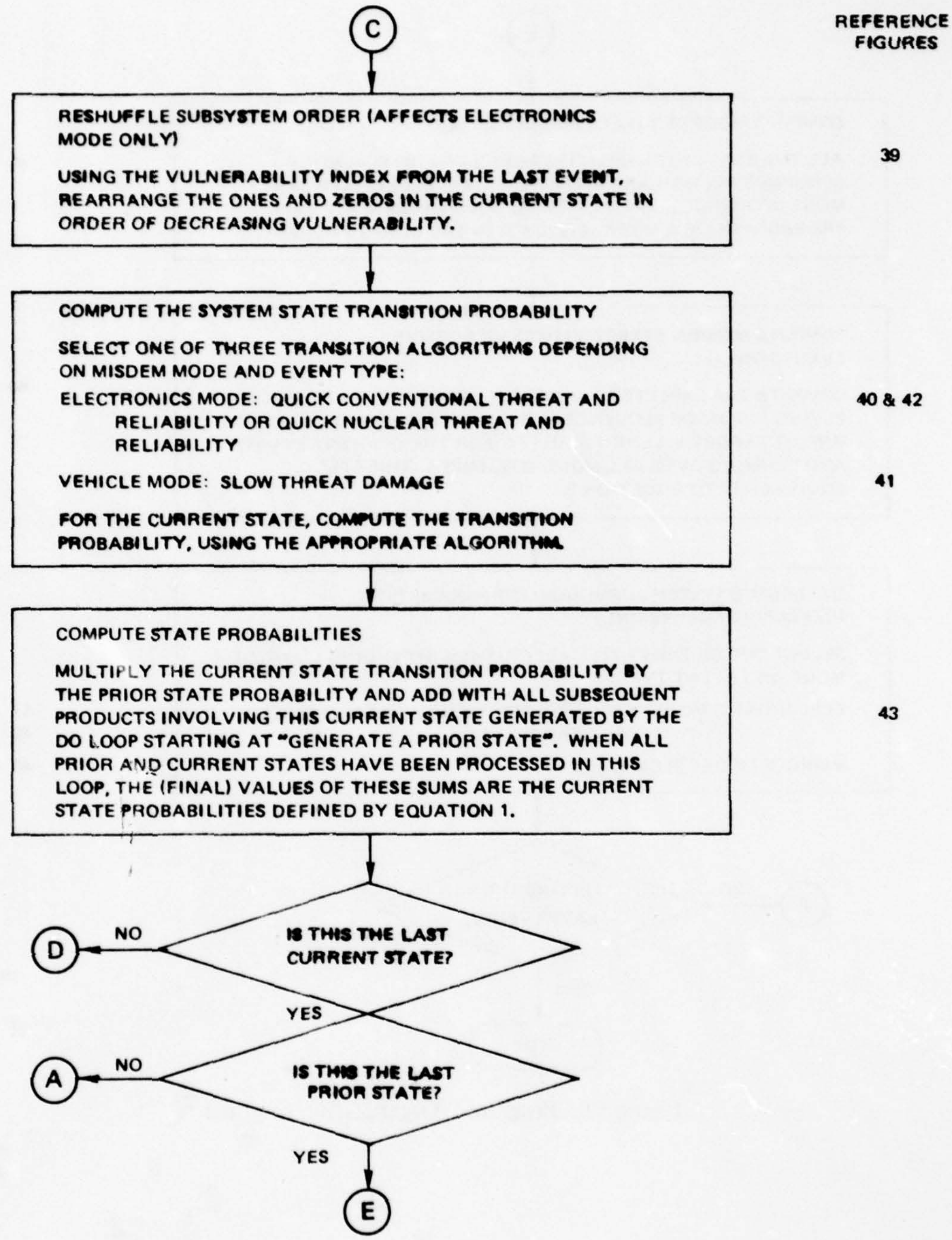


Figure 20. Program 2 Organization (contd.).

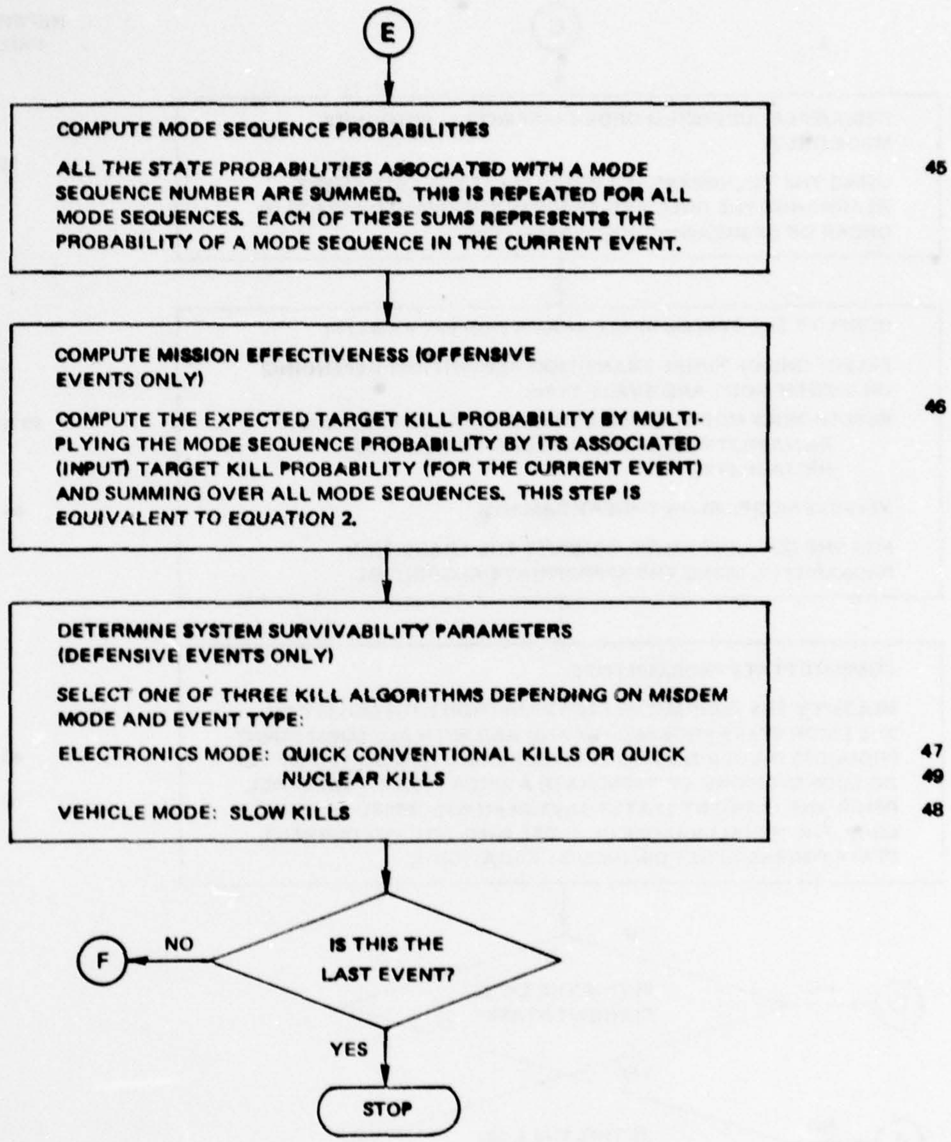


Figure 20. Program 2 Organization (contd.).

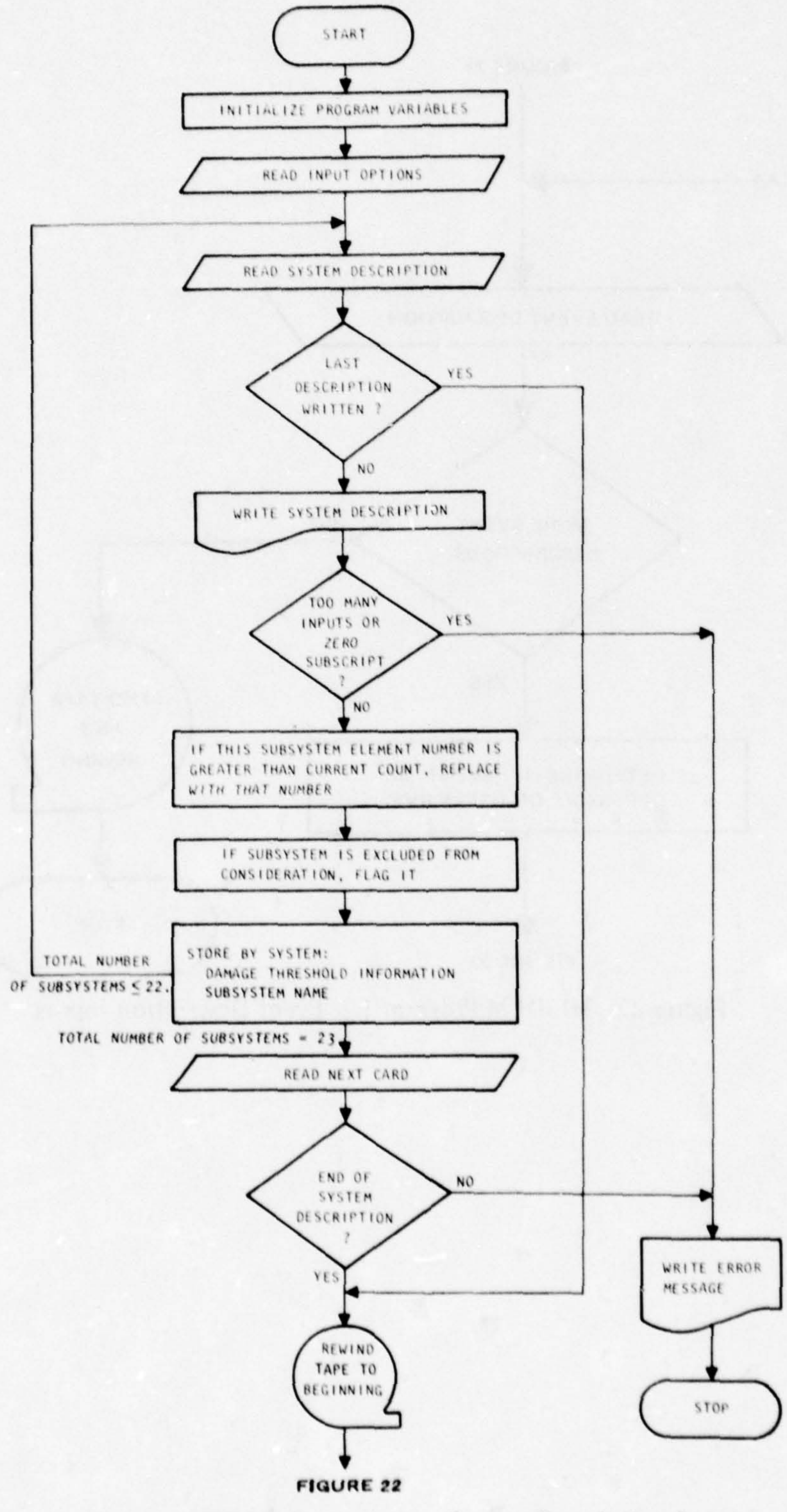


Figure 21. MISDEM Program 1 – System Description Inputs.

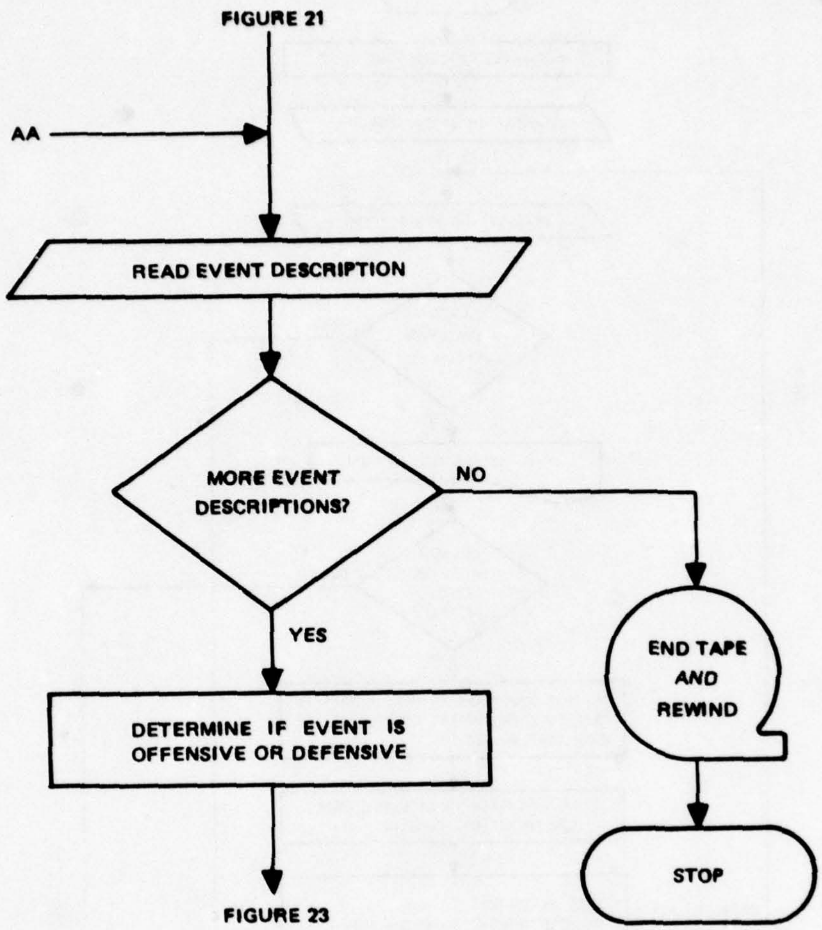


Figure 22. MISDEM Program 1 – Event Description Inputs.

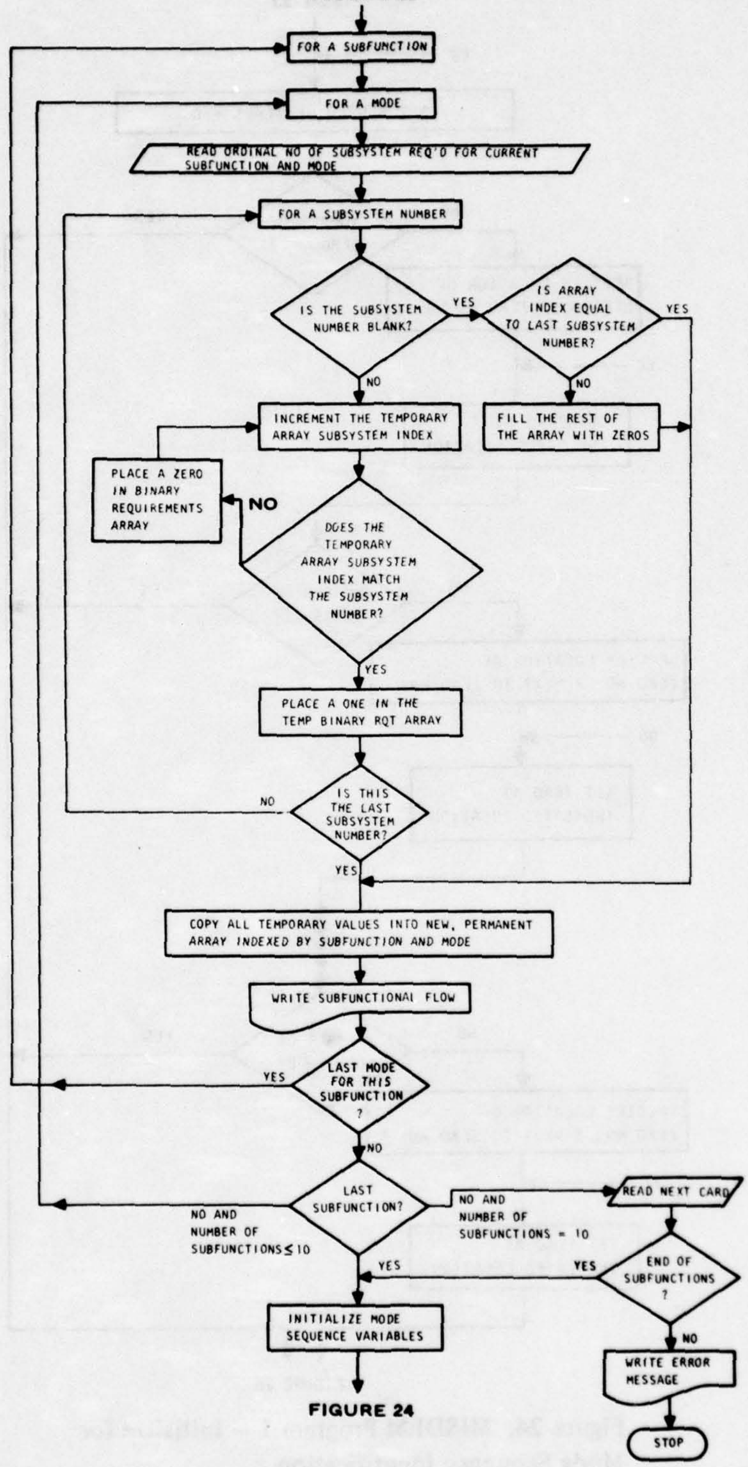


Figure 23. MISDEM Program 1 – Read and Write Subfunctional Flow.

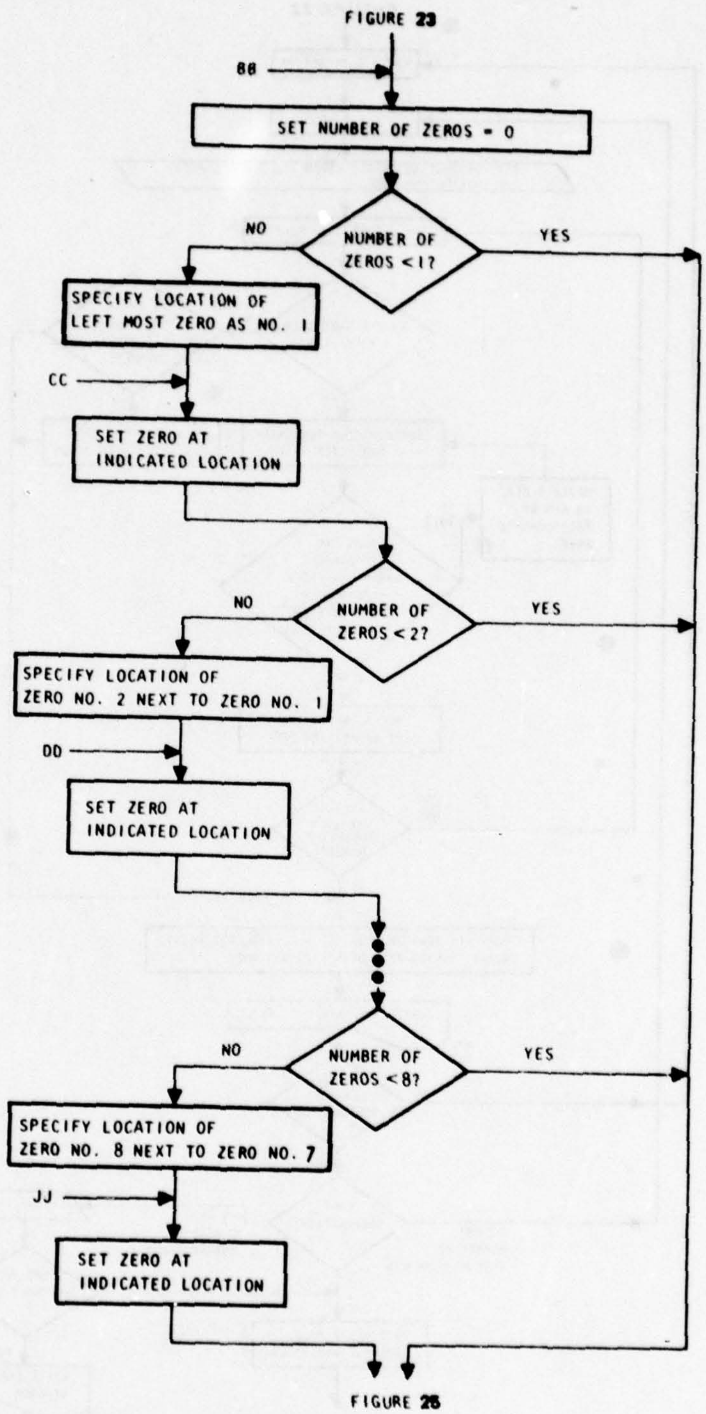


Figure 24. MISDEM Program 1 – Initialize for Mode Sequence Identification

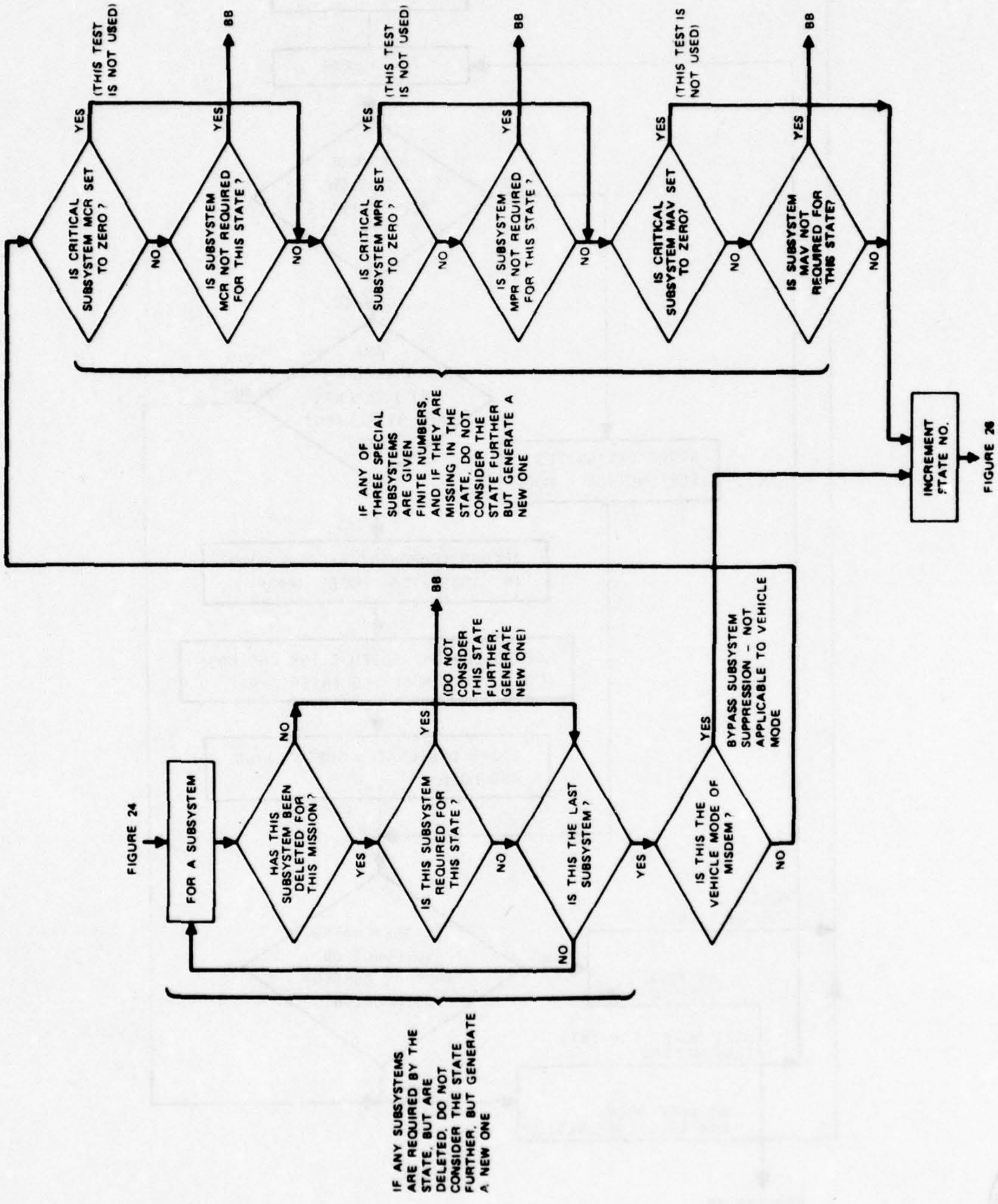


Figure 25. MISDEM Program 1 – Test for Suppressed Subsystems.

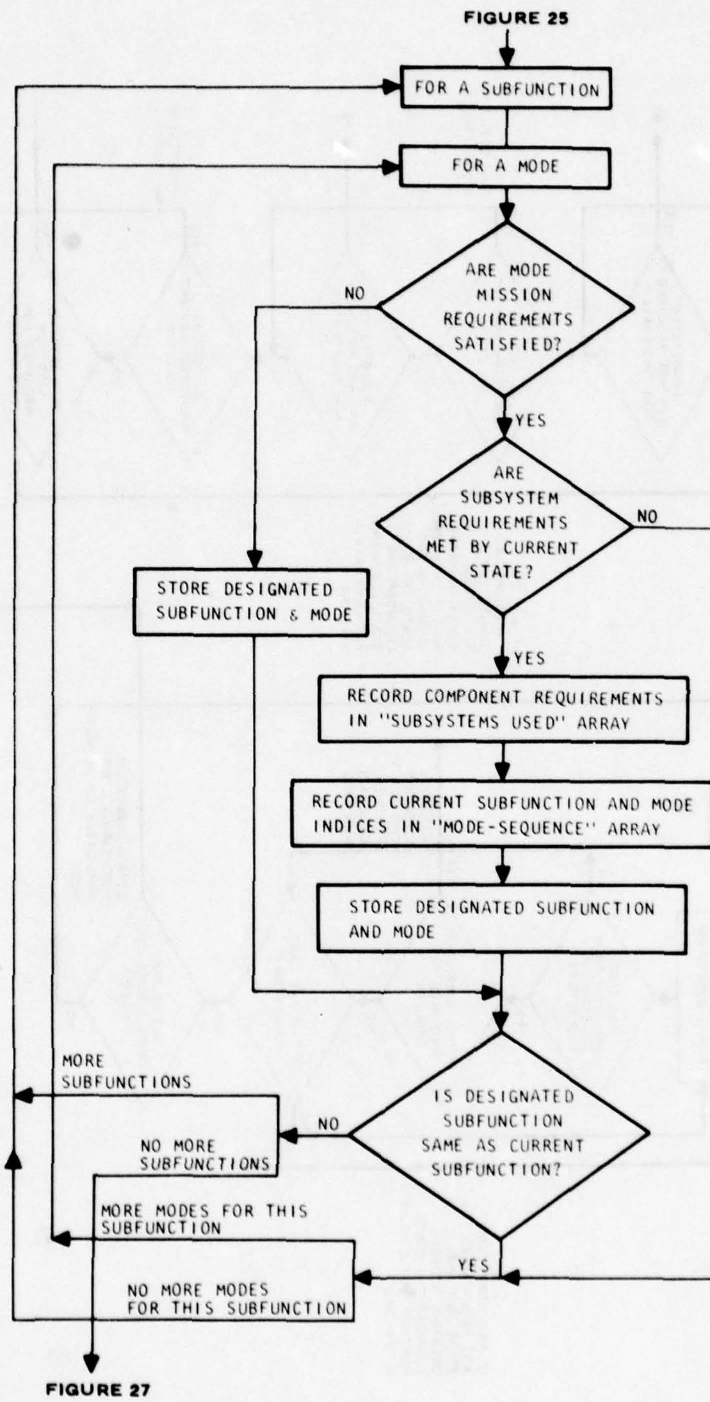


Figure 26. MISDEM Program 1 – Define Mode Sequence and Subsystems Used.

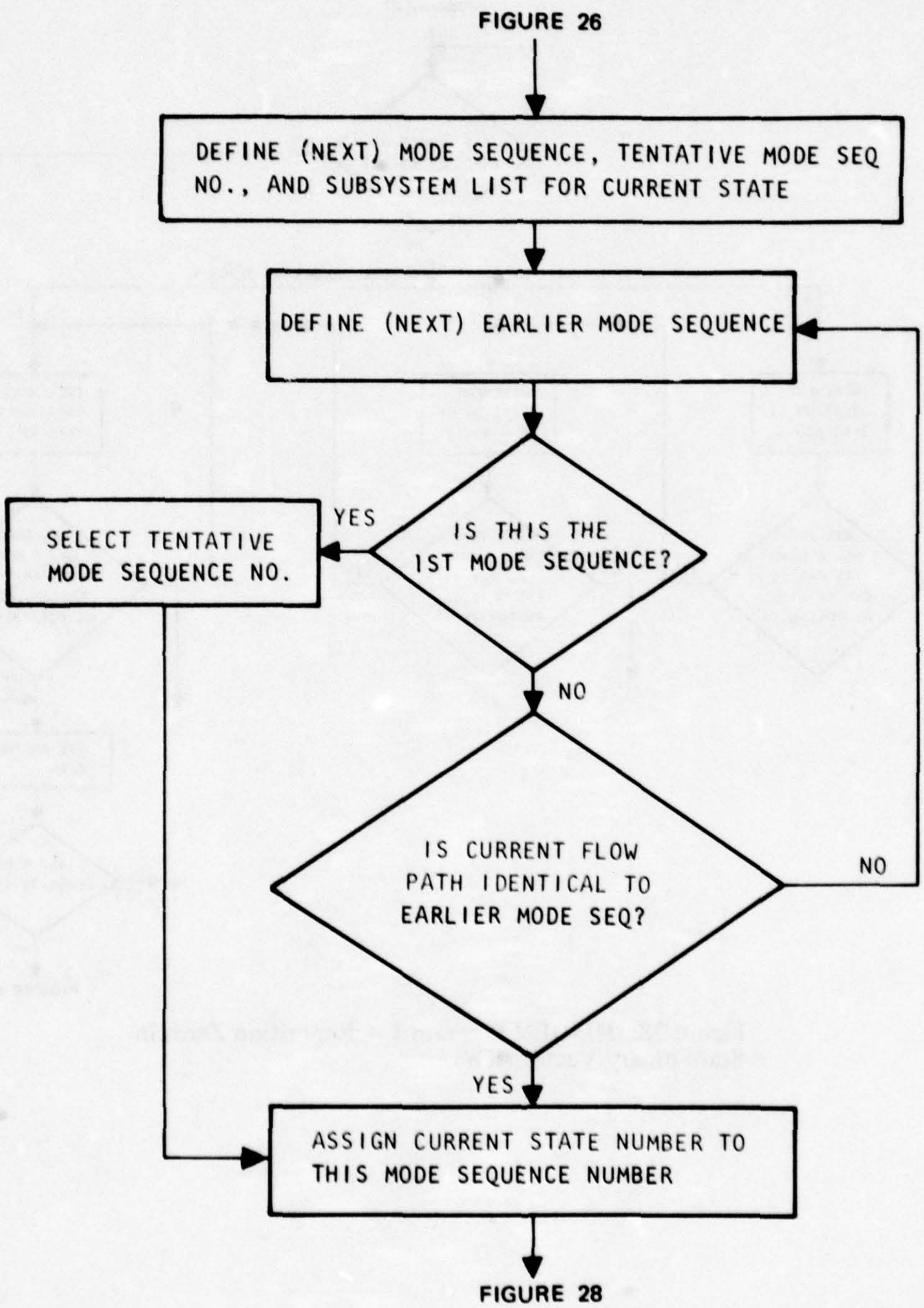


Figure 27. MISDEM Program 1 – Assign Mode Sequence Numbers.

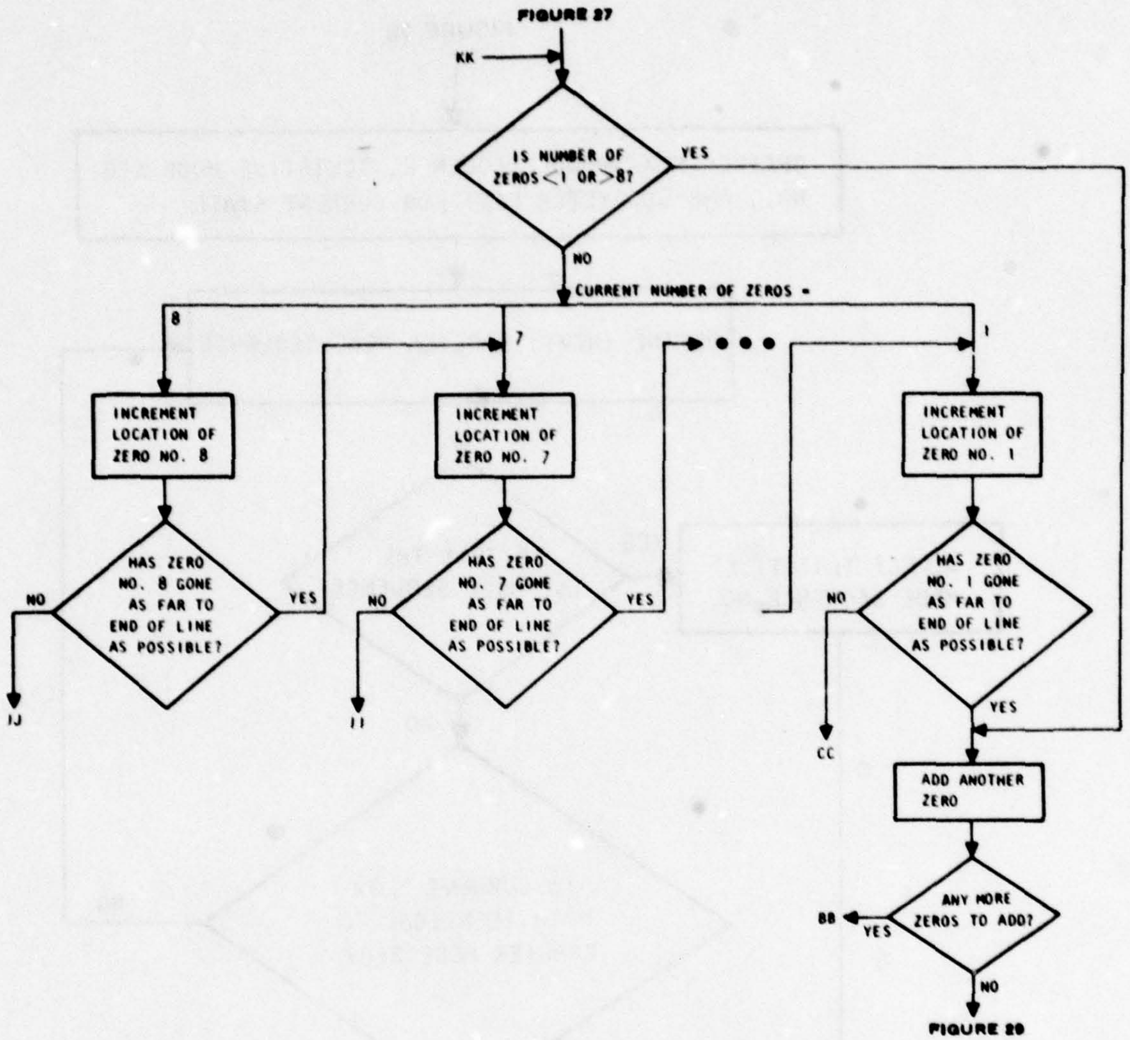


Figure 28. MISDEM Program 1 – Reposition Zeros in State Binary Vector (KW).

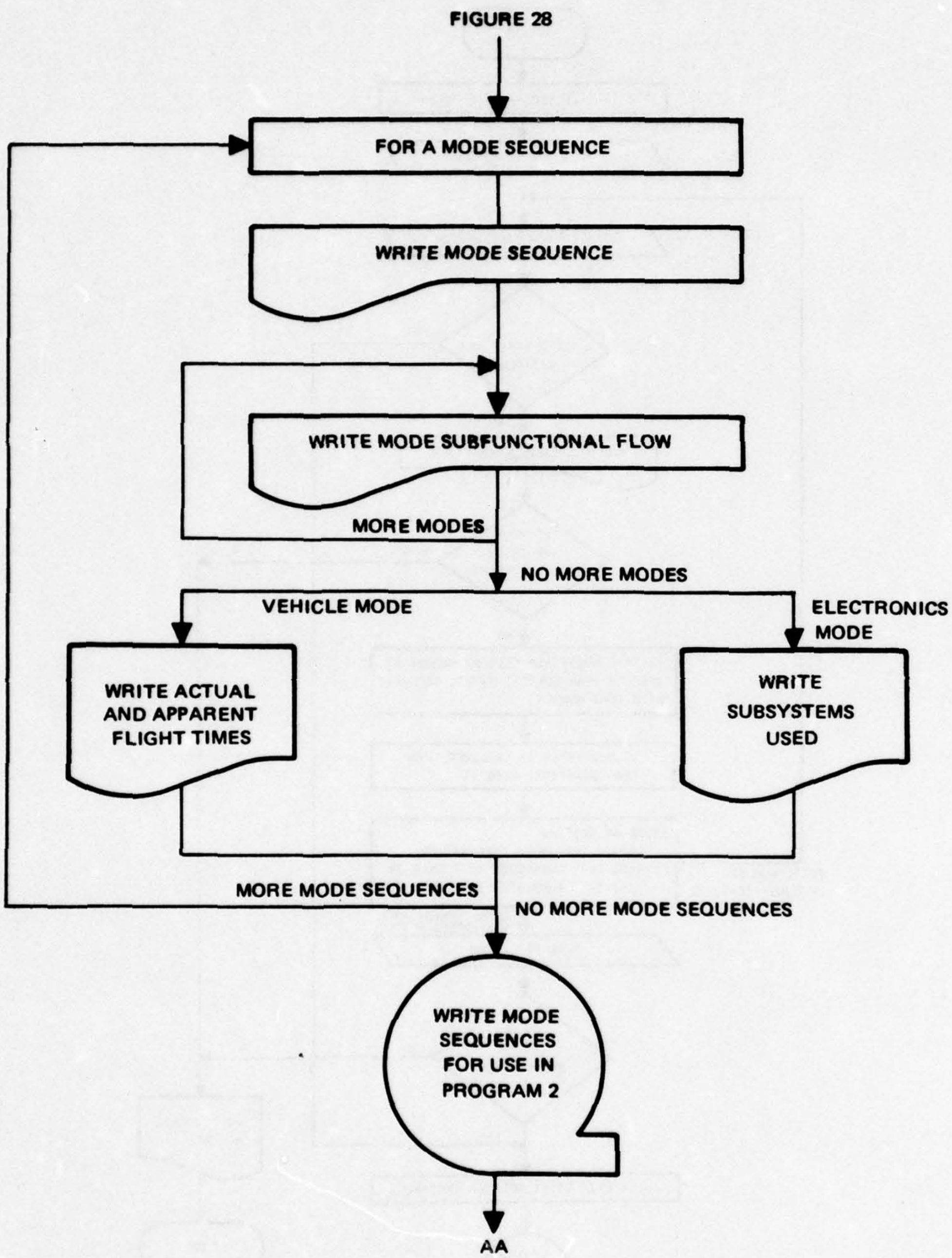


Figure 29. MISDEM Program 1 – Write Mode Sequence and Subsystems Used.

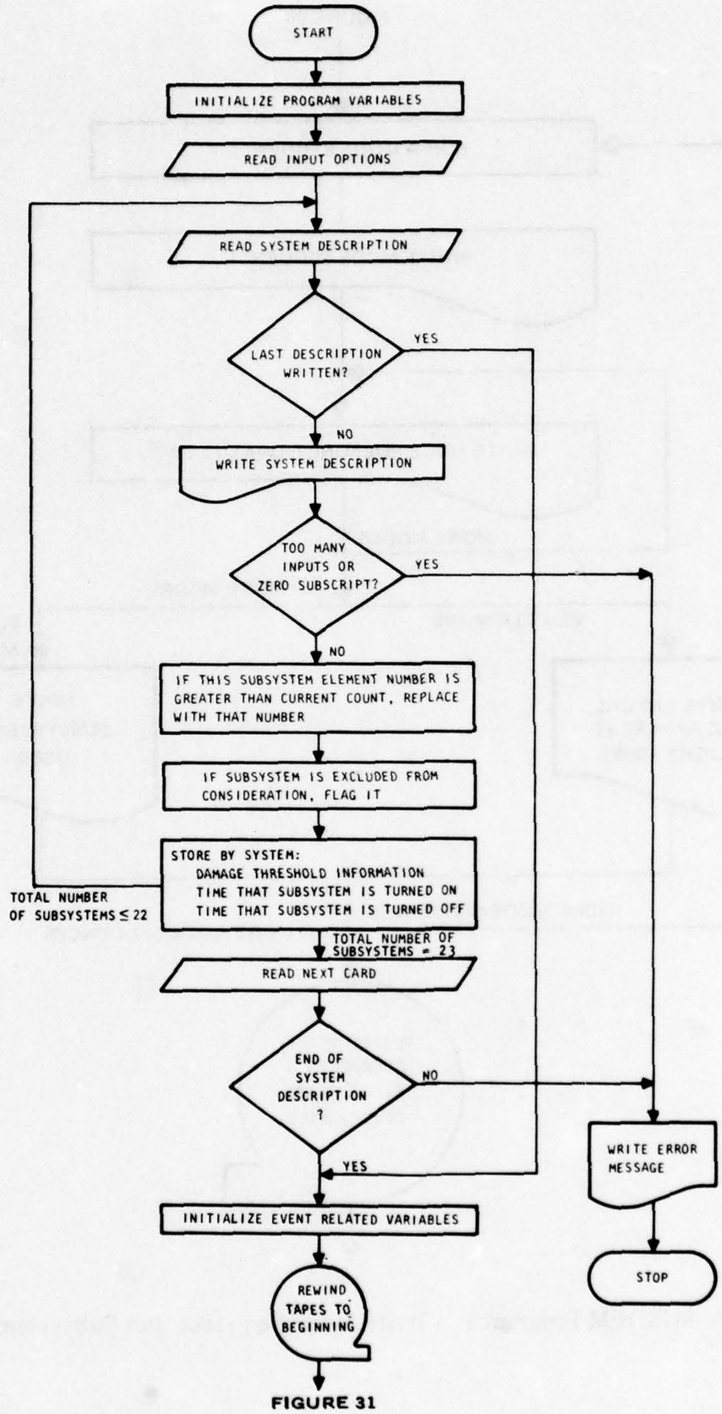


Figure 30. MISDEM Program 2 – System Description Inputs.

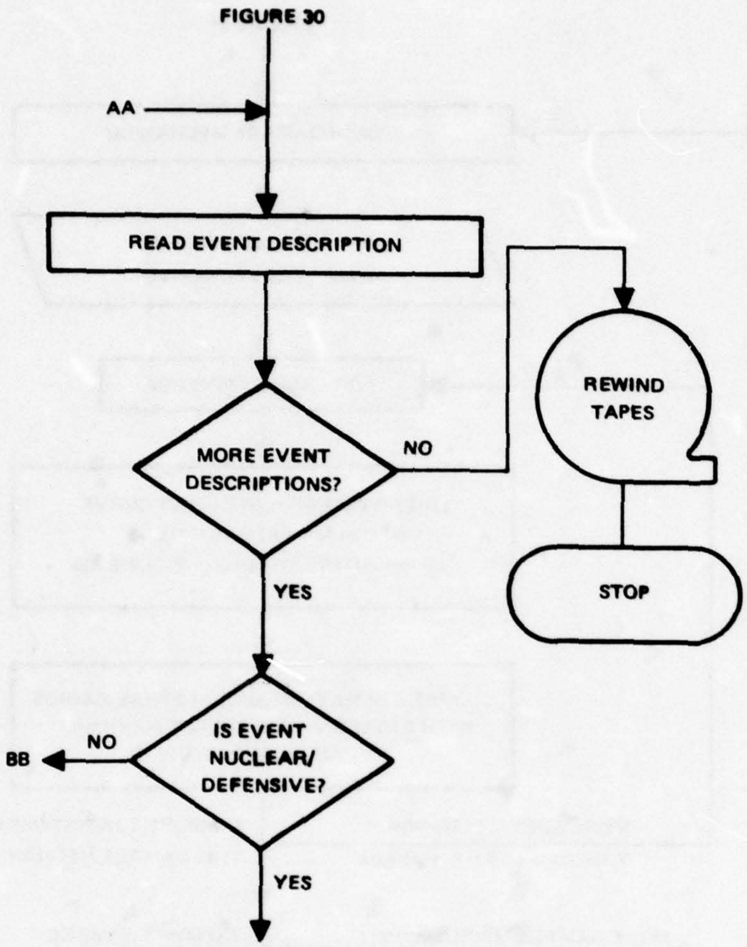


Figure 31. MISDEM Program 2 – Event Description Inputs.

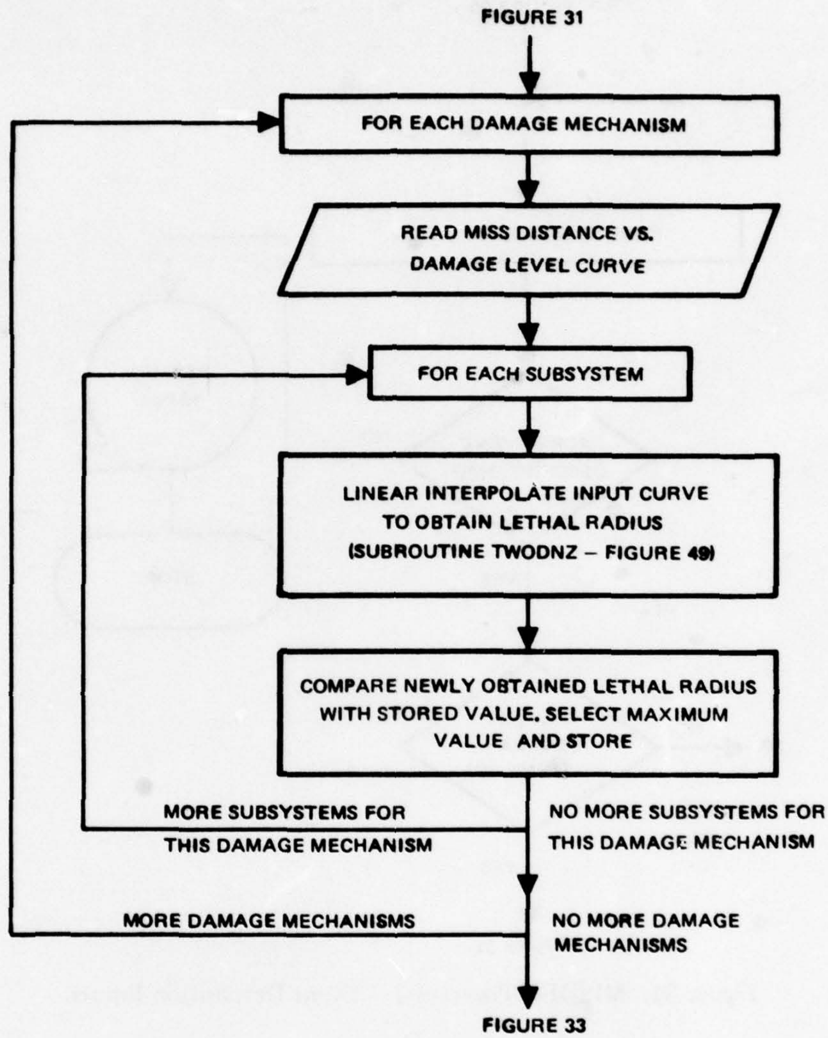


Figure 32. MISDEM Program 2 – Compute Lethal Radius.

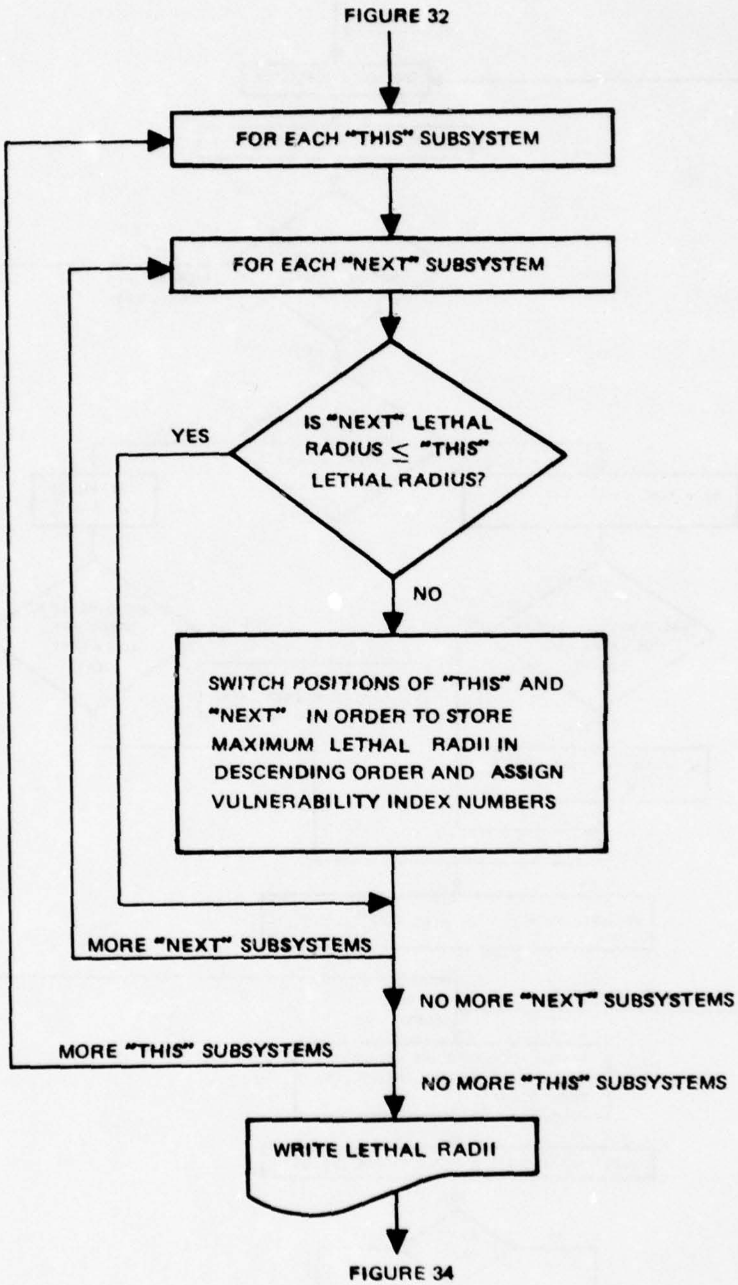


Figure 33. MISDEM Program 2 – Assign Vulnerability Index Numbers.

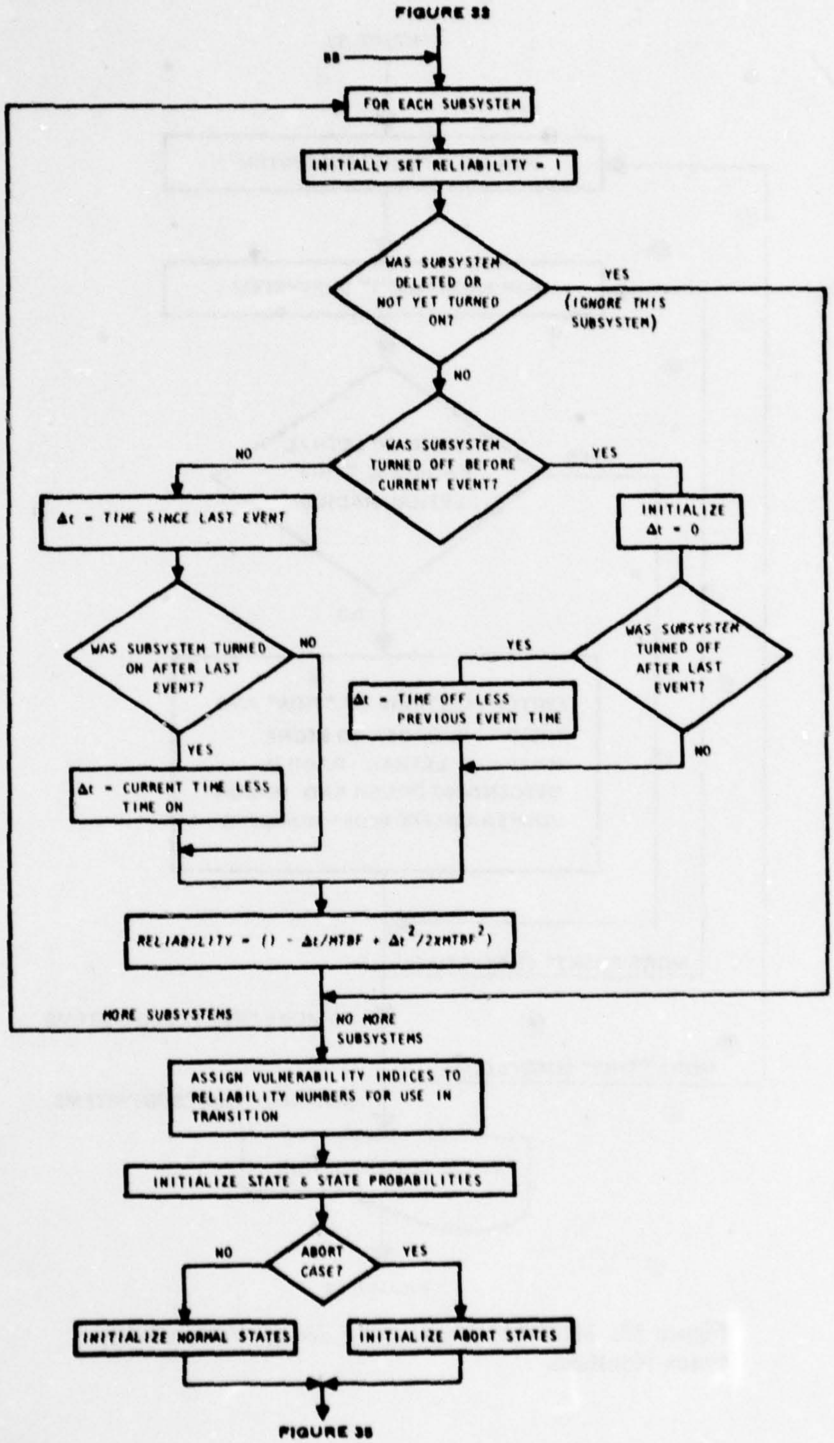


Figure 34. MISDEM Program 2 – Compute the Reliability of Every Subsystem and Reshuffle Reliability Order.

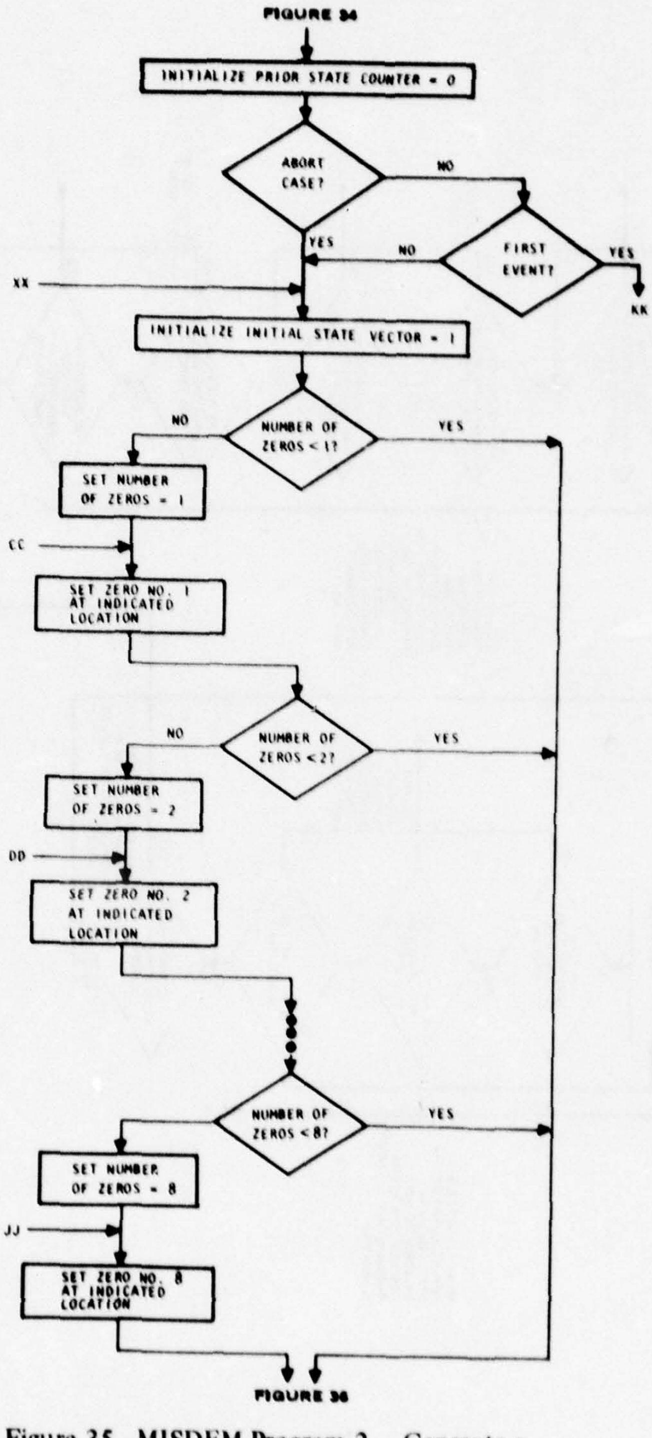


Figure 35. MISDEM Program 2 – Generate a Prior State of the System.

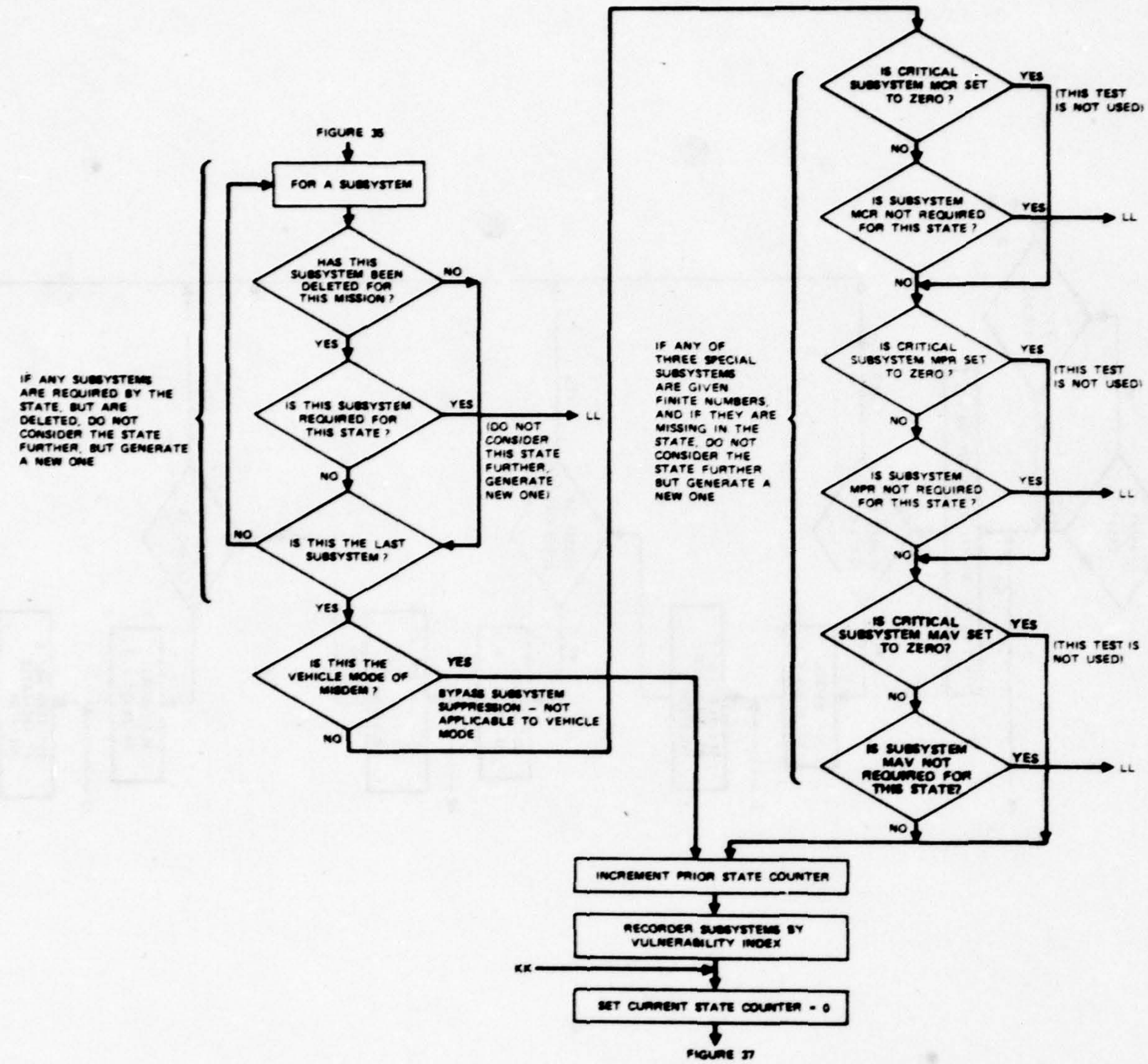


Figure 36. MISDEM Program 2 – Test for Suppressed Systems in Prior State and Reorder Subsystems.

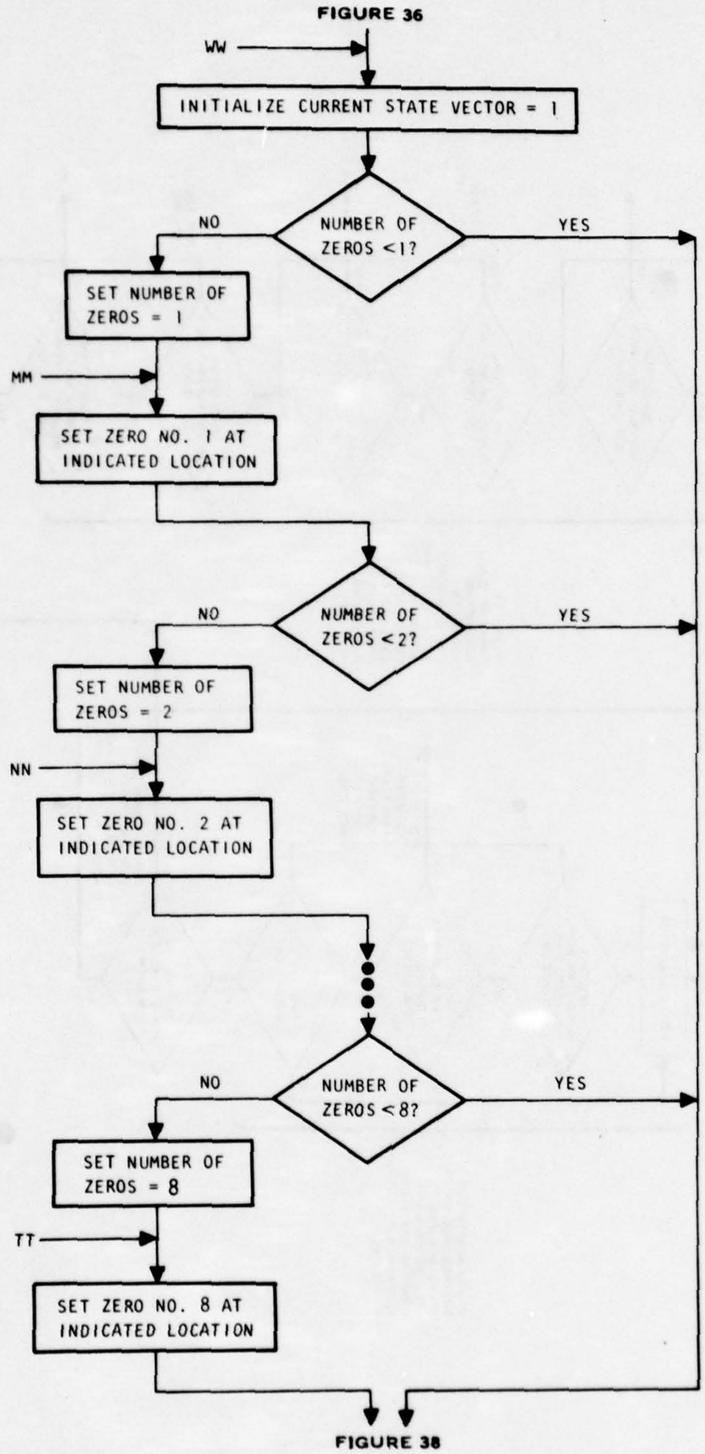


Figure 37. MISDEM Program 2 – Generate Current State of the System.

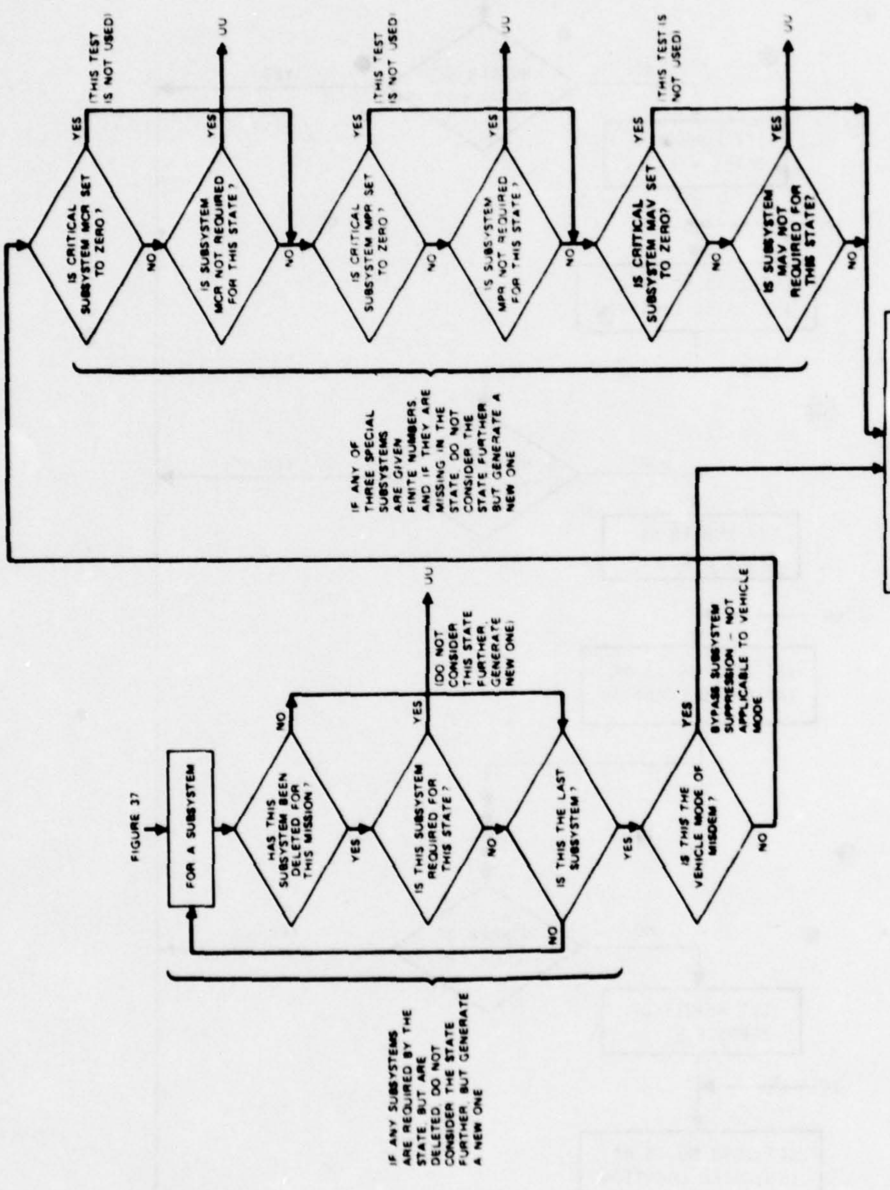


Figure 38. MISDEM Program 2 – Test for Suppressed Subsystems in Current State and Reorder Subsystems.

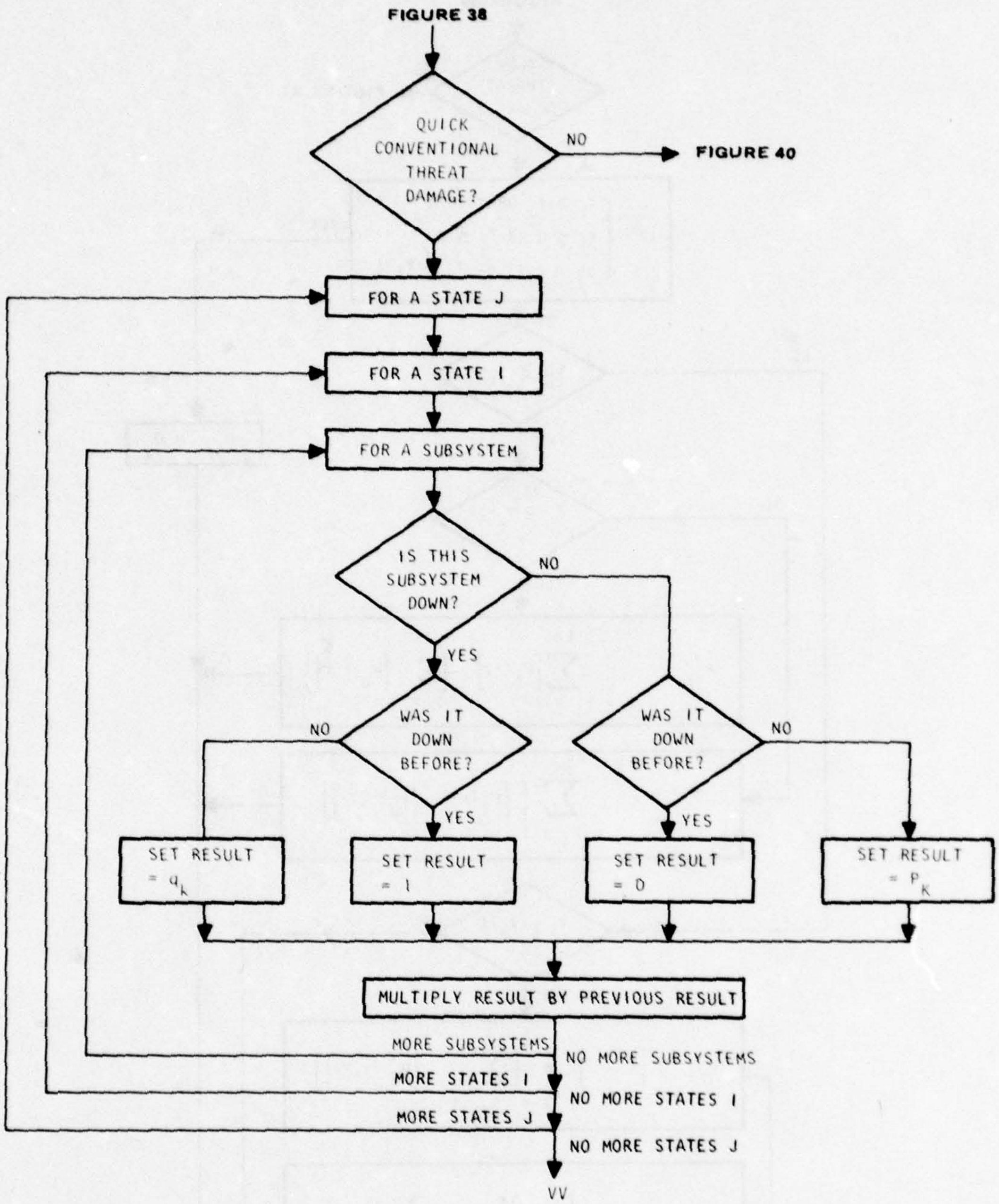


Figure 39. MISDEM Program 2 – Quick Conventional Damage.

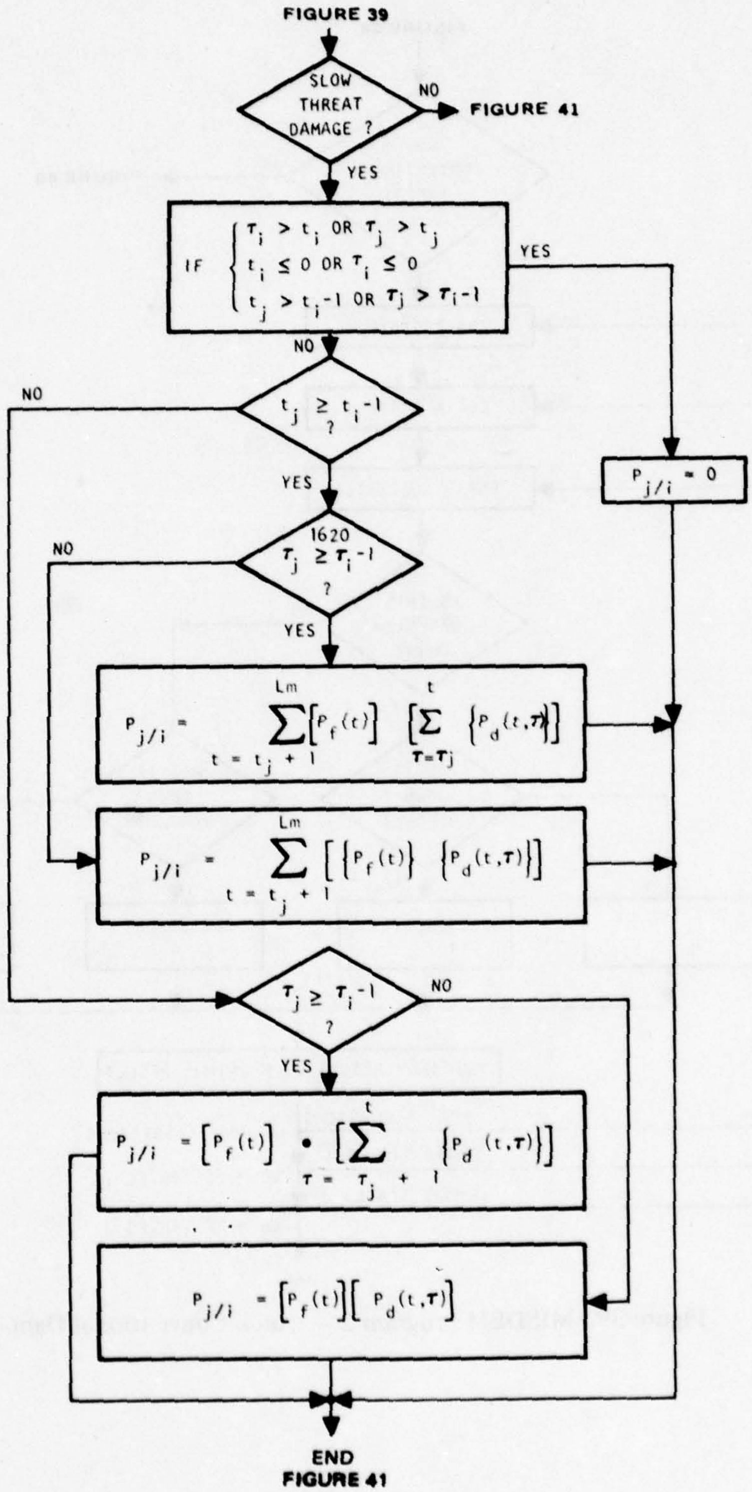


Figure 40. MISDEM Program 2 – Slow Threat Damage.

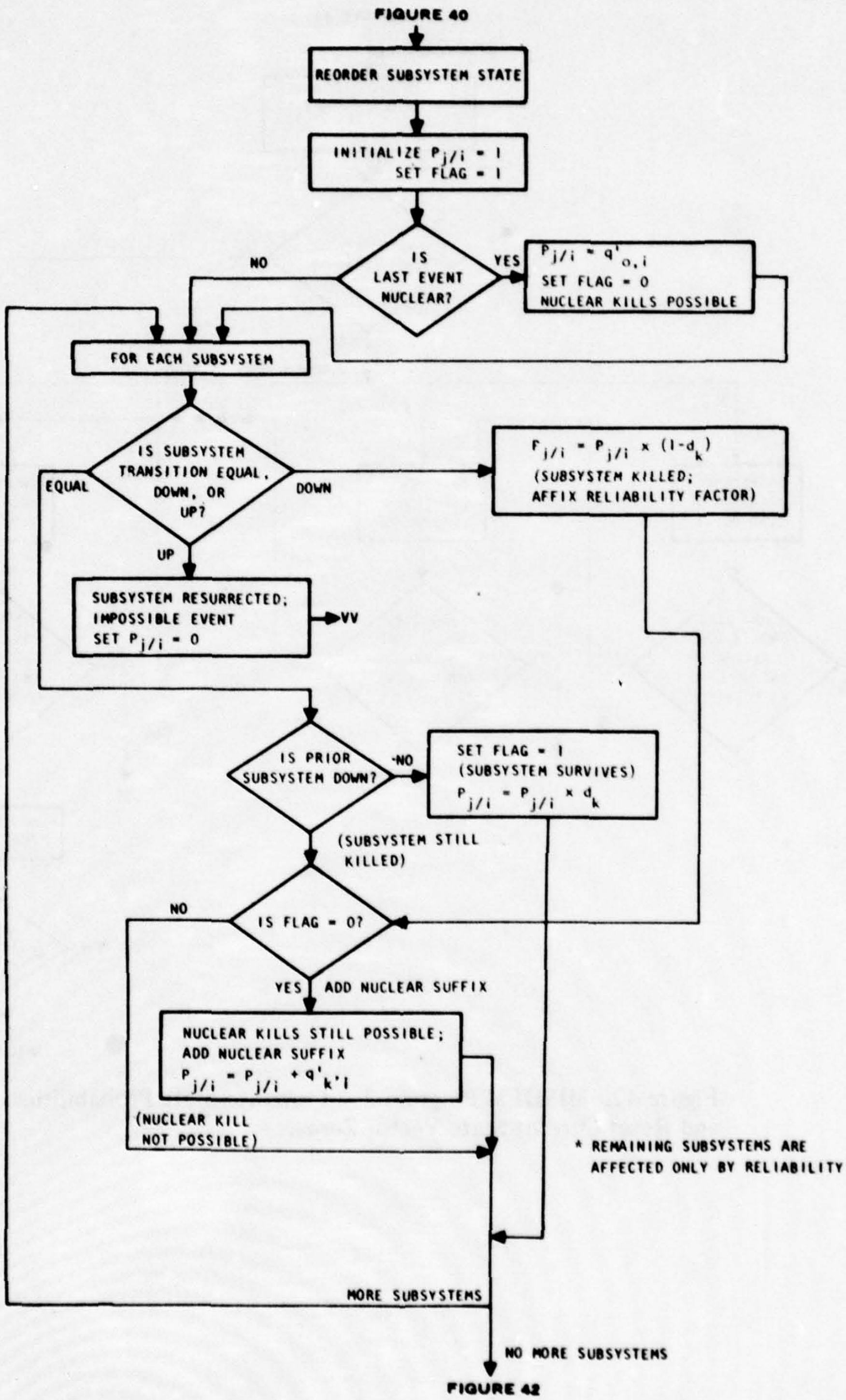


Figure 41. MISDEM Program 2 – Quick Nuclear Damage and Reliability and Non-Defensive Transitions.

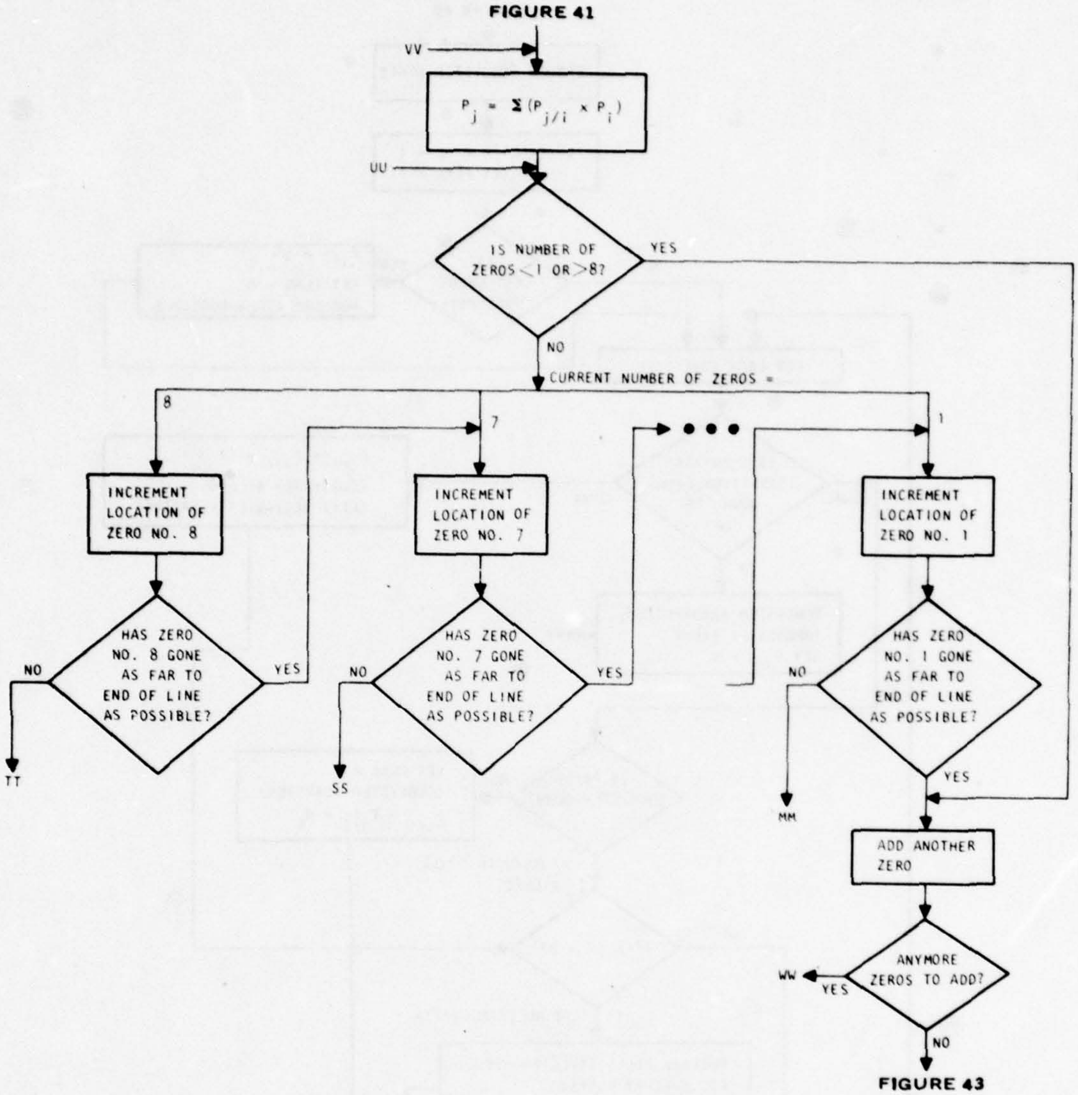


Figure 42. MISDEM Program 2 – Compute State Probabilities and Reset Current State Vector Zeros.

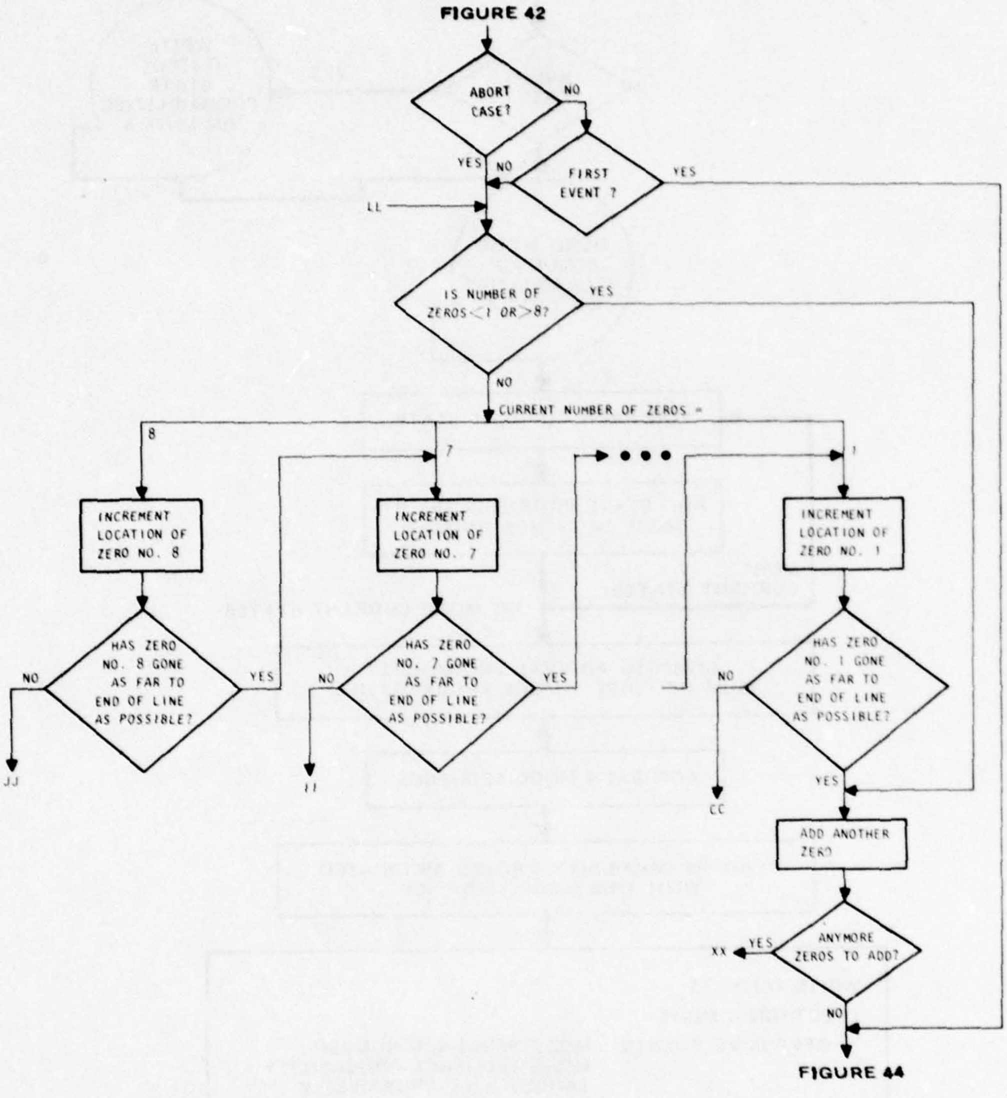


Figure 43. MISDEM Program 2 – Reset Prior State Vector Zeros.

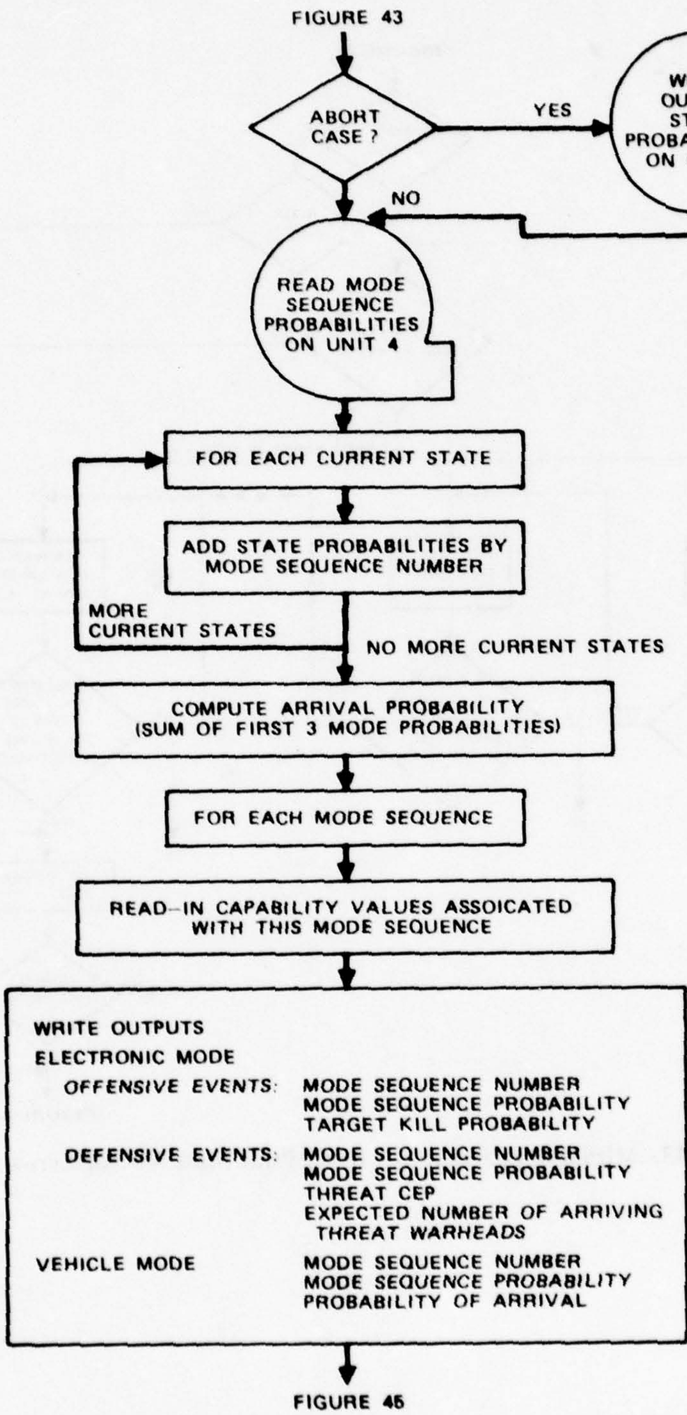


Figure 44. MISDEM Program 2 – Compute and Write Mode Sequence Probabilities.

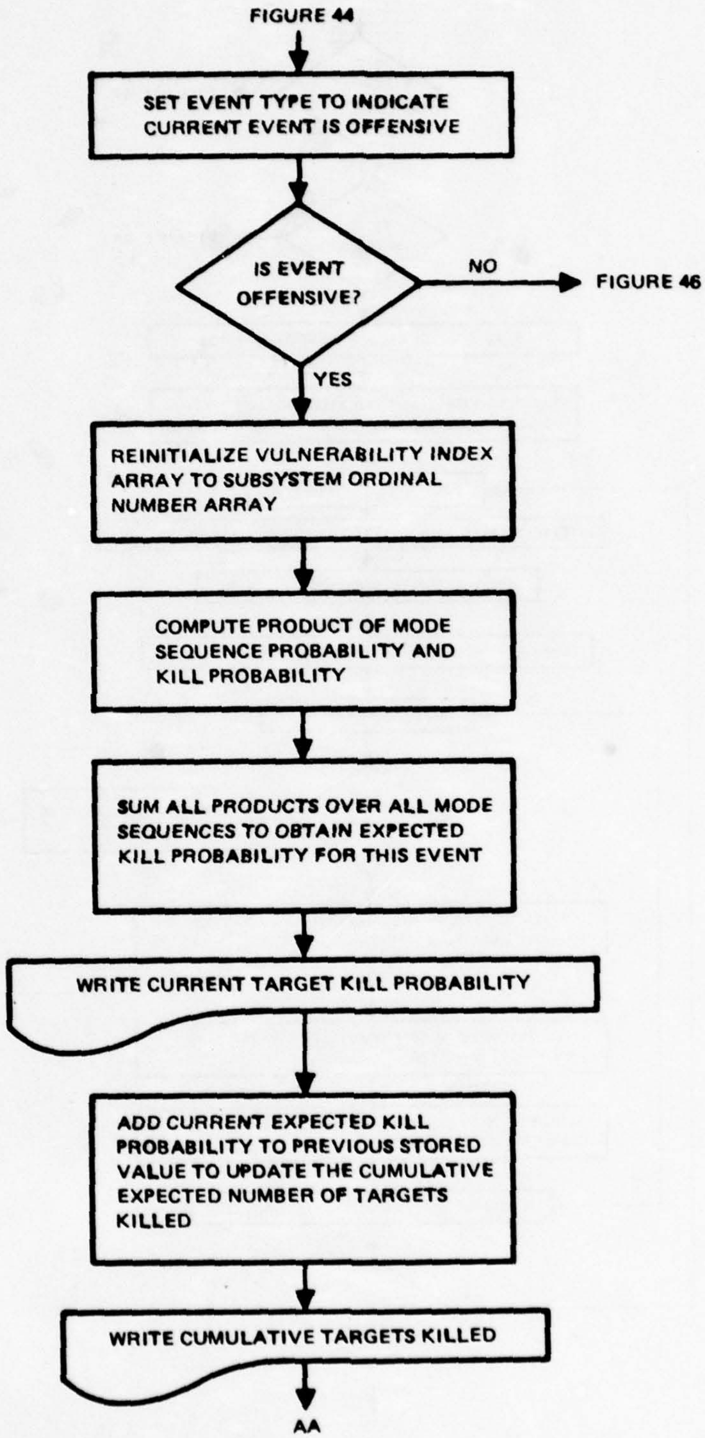


Figure 45. MISDEM Program 2 – Compute Offensive Events Mission Effectiveness.

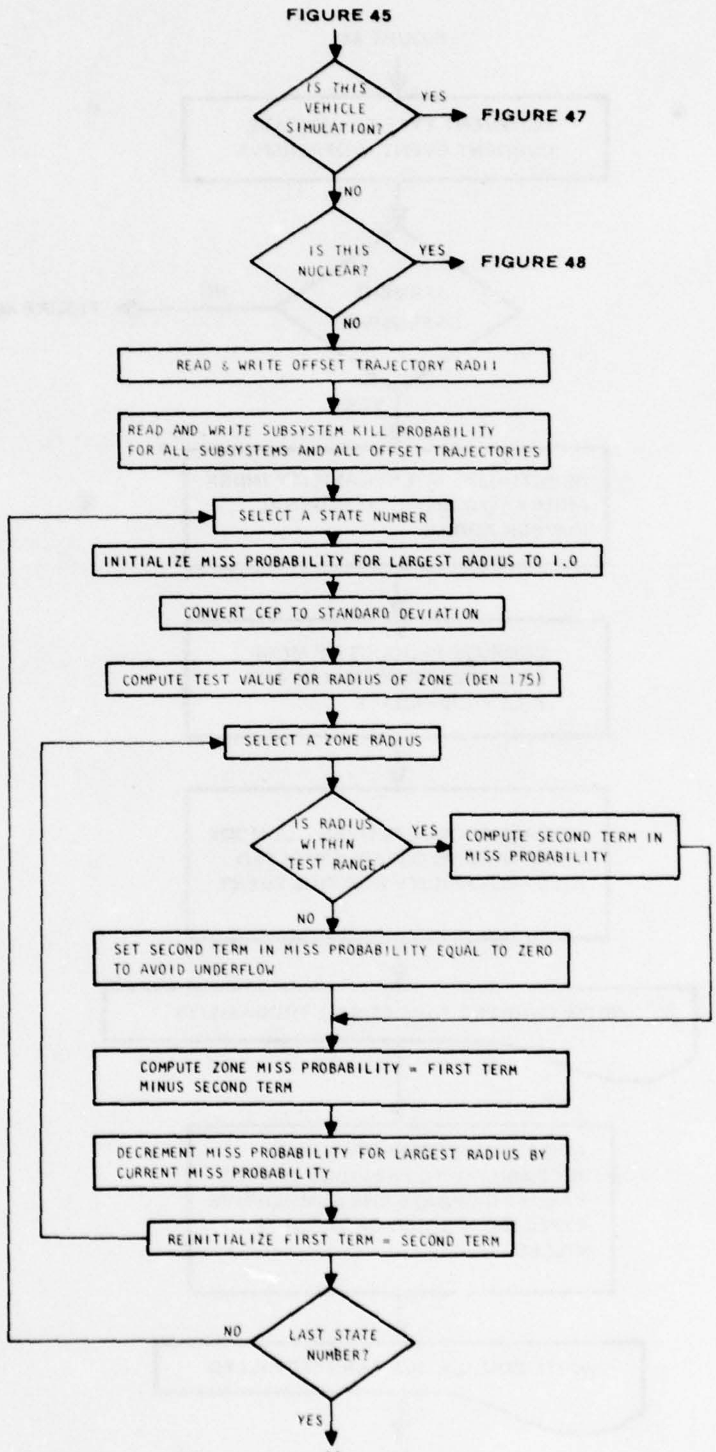


Figure 46. MISDEM Program 2 – Quick Conventional Kills.

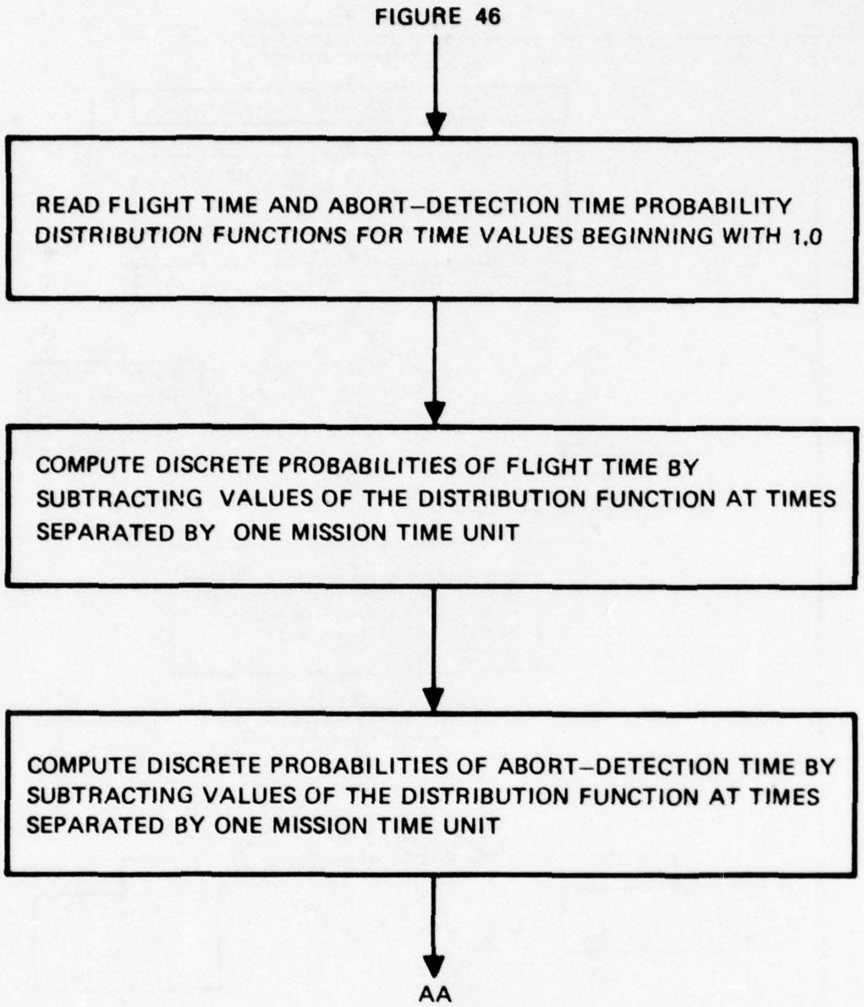


Figure 47. MISDEM Program 2 – Slow Kills.

FIGURE 46

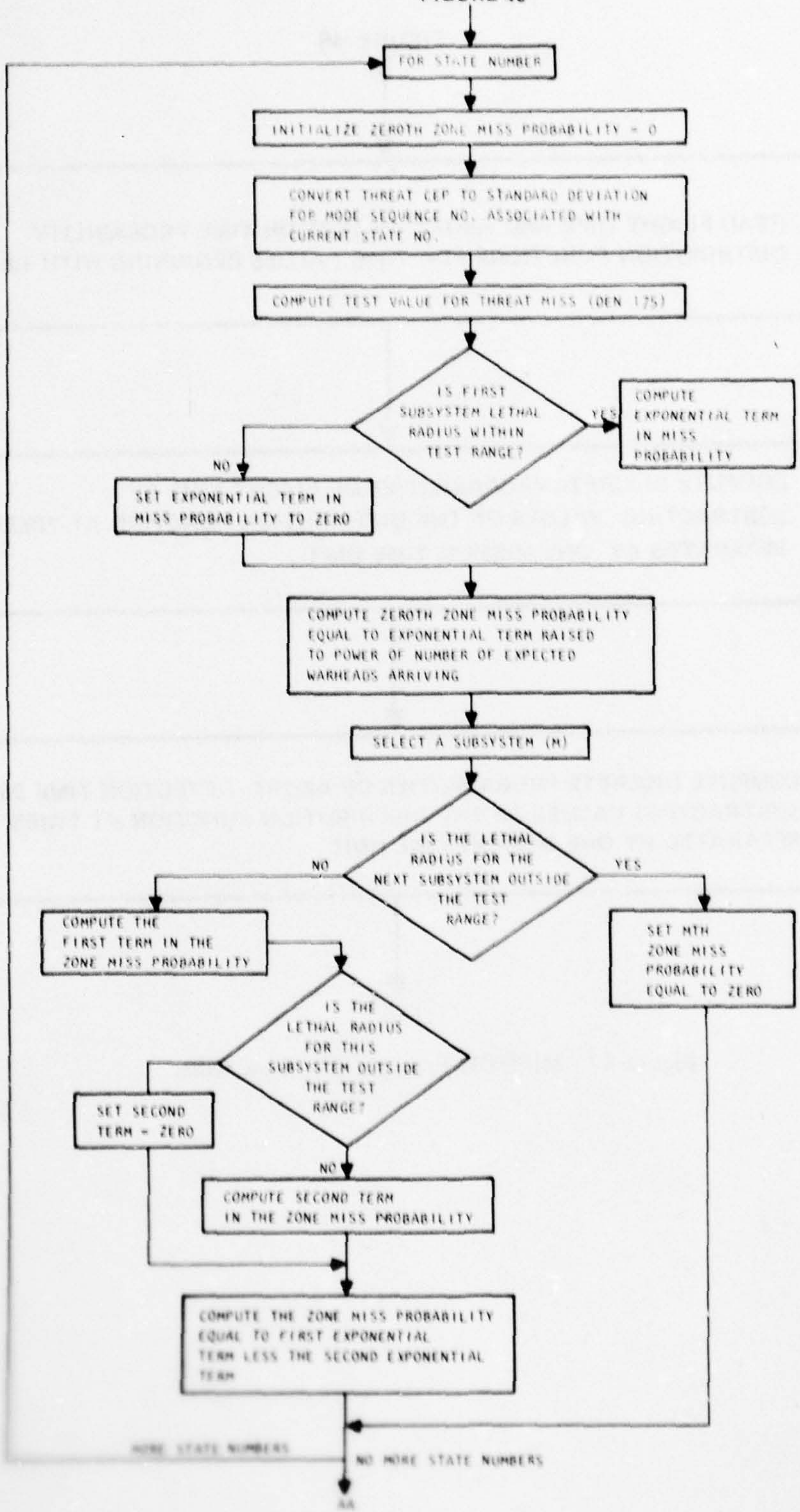


Figure 48. MISDEM Program 2 – Quick Nuclear Kills.

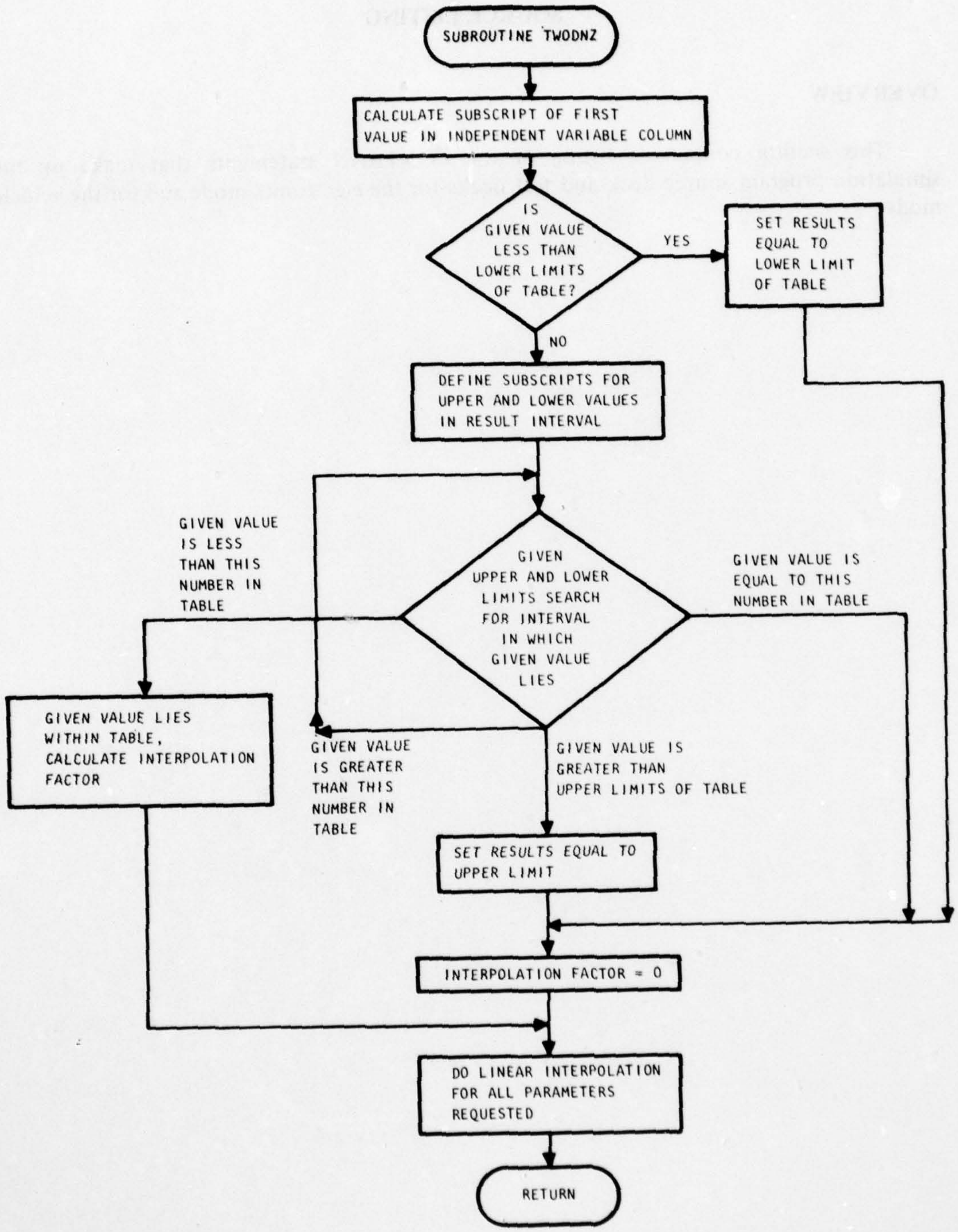


Figure 49. MISDEM Program 2-Linear Interpolation Routine (Subroutine TWODNZ).

SOURCE LISTING

OVERVIEW

This section contains a listing of the FORTRAN statements that make up the simulation program source deck and test decks for the electronics mode and for the vehicle mode.

SIMULATION SOURCE DECK

```
C ***********************************************************************
C
C         MISSION DAMAGE EFFECTIVENESS MODEL - PROGRAM ONE
C                         MISDEM - PGM 1
C
C ***********************************************************************
      DIMENSION CN(3),CNAME(8,23),DEF(3),DNAME(8),FNAME(10,27,10),
     1  HDMT(4),JES(10,27,25),JU(23,25),KS(10,27),KU(23),KW(23),LEE(10),
     2  LF(10),LLF(27),LLQ(10,27),LMA(10,27,23),LMAT(10,27,23),LMATT(23)
     3  ,LQ(10,27),MD(10,27),MI(23),MMQ(10,27),MQ(10,27),OFF(3),WEAPN(7)
      INTEGER*2 IJN(256)
      LOGICAL MD,MDT
      EQUIVALENCE (LMA(1,1,1),LMAT(1,1,1))
      DATA BLK/4H    /,OFF/4HOFFE,4HNSIV,4HE    /,DEF/4HDEFE,4HNSIV,
     *4HE    /,OUT/2HNO/
C
C1        INITIALIZE
C
      L1=27
      LCN8=8
      L40=10
      LCEI=23
      L72=7
      INCEI=0
      DO 20 I=1,LCEI
      MI(I)=1
      DO 20 J=1,LCN8
      CNAME(J,I)=BLK
 20   CONTINUE
      READ(5,950)NZT,MCR,MPR,MAV
C
C2        READ AND WRITE SYSTEM DESCRIPTION
C
      WRITE(6,960)
      WRITE(6,970)

      DO 40 I=1,LCEI
      READ(5,980) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),ONOFF
      IF(M.EQ.999)GO TO 60
      WRITE(6,990)M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),ONOFF
      IF(M.GT.LCEI.OR.M.LT.1) GO TO 50
      IF(M.GT.INCEI)INCEI=M
      IF(ONOFF.EQ.OUT)MI(I)=0
      DO 40 J=1,LCN8
      CNAME(J,I)=DNAME(J)
   40 CONTINUE
      READ(5,980) M
      IF(M.EQ.999) GO TO 60
 50   WRITE(6,1000)
      STOP
   60 CONTINUE
      REWIND 4
   70 CONTINUE
```

```
C
C3        READ AND WRITE EVENT DESCRIPTION
C
      READ(5,1010,END=840) IEVENT,T2,NC,(WEAPN(I),I=1,7),MDT
      WRITE(6,1270)
      DO 71 N=1,3
      CN(N)=OFF(N)
      IF(NC.NE.0) CN(N)=DEF(N)
   71 CONTINUE
      WRITE(6,1020) IEVENT, (CN(N),N=1,3), T2, (WEAPN(I),I=1,7), MDT
      IF(NC.NE.7) WRITE(6,1060)
         IF(NC.EQ.7) WRITE(6,1065)
C
C4        READ SUBFUNCTIONAL FLOW FOR EVENT AND GENERATE SUBSYSTEM
C                     REQUIREMENTS VECTOR
C
      DO 250 I=1,L40
      DO 240 K=1,L1
      MD(I,K)=.FALSE.
      READ(5,1070)LF(I),LLF(K),LQ(I,K),LLQ(I,K),(FNAME(I,K,J),J=1,10),
     *              MQ(I,K),MMQ(I,K),MD(I,K),(LMAT(I,K,J),J=1,INCEI)
      L=0
      DO 90 J=1,INCEI

       IF(LMAT(I,K,J).EQ.0)GO TO 100
   80  L=L+1
       IF(LMAT(I,K,J).EQ.L)GO TO 90
       LMATT(L)=0
       GO TO 80
   90  LMATT(L)=1
       GO TO 140
  100  J=L+1
       IF(J.GT.INCEI)GO TO 140
       DO 110 L=J,INCEI
  110  LMATT(L)=0
  140  DO 150 L=1,INCEI
  150  LMA(I,K,L)    =LMATT(L)
C
C5         WRITE SUBFUNCTIONAL FLOW
C
       IF(NC.NE.7)
     1WRITE(6,1080)LF(I),LLF(K),LQ(I,K),LLQ(I,K),(FNAME(I,K,J),J=1,10),
     *              MQ(I,K),MMQ(I,K),MD(I,K),(LMAT(I,K,J),J=1,INCEI)
       IF(NC.EQ.7) WRITE(6,1085) LF(I),LLF(K),LQ(I,K),LLQ(I,K),
     *   (FNAME(I,K,J),J=1,10),(LMAT(I,K,J),J=1,INCEI)
       IF(LLF(K)    .EQ.99) GO TO 230
       IF(LF(I)    .EQ.999) GO TO 260
       GO TO 240
  230  LEE(I)=K-1
       GO TO 250
  240  CONTINUE
  250  CONTINUE
       I=L40+1
       READ(5,1070)M
       IF(M.EQ.999) GO TO 260
       WRITE(6,1090)
       STOP
```

```
C
C6        INITIALIZE FOR MODE SEQUENCE IDENTIFICATION
C
 260  LE=I-1
      LQSVL=0
      LLQS=0
      NZ=0
      DO 290 J=1,L40
      DO 290 K=1,L1
      KS(J,K)=0
      DO 290 I=1,L72
      JES(J,K,I)=0
 290  CONTINUE
      DO 300 I=1,LCEI
      KW(I)=0
 300  CONTINUE
      DO 310 JCOUNT=1,256
      IJN(JCOUNT)=0
 310  CONTINUE
      JCOUNT=0
      JCAP=0
C
C7        GENERATE FINAL STATE OF THE SYSTEM - BINARY VECTOR KW
C
 330  CONTINUE
      DO 340 I=1,INCEI
      KW(I)=1
 340  CONTINUE
      IF(NZ.LT.1)GO TO 400
      LL1=1
 350  KW(LL1)=0
      IF(NZ.LT.2)GO TO 400
      LL2=LL1+1
 360  KW(LL2)=0
      IF(NZ.LT.3)GO TO 400
      LL3=LL2+1
 370  KW(LL3)=0
      IF(NZ.LT.4)GO TO 400
      LL4=LL3+1
 380  KW(LL4)=0
      IF(NZ.LT.5)GO TO 400
      LL5=LL4+1
 390  KW(LL5)=0
      IF(NZ.LT.6) GO TO 400
      LL6=LL5+1
  391 KW(LL6)=0
      IF(NZ.LT.7) GO TO 400
      LL7=LL6+1
  392 KW(LL7)=0
      IF(NZ.LT.8) GO TO 400
      LL8=LL7+1
  393 KW(LL8)=0
 400  CONTINUE
```

```
C
C8        TEST FOR SUPPRESSED SUBSYSTEMS
C
      DO 410 MM=1,INCEI
      IF(MI(MM).NE.0)GO TO 410
      IF(KW(MM).EQ.1)GO TO 720
  410 CONTINUE
      IF(NC.EQ.7)GO TO 405
      IF(MCR.EQ.0)GO TO 402
      IF(KW(MCR).EQ.0)GO TO 720
  402 IF(MPR.EQ.0)GO TO 404
      IF(KW(MPR).EQ.0)GO TO 720
  404 IF(MAV.EQ.0)GO TO 405
      IF(KW(MAV).EQ.0)GO TO 720
  405 CONTINUE
      JCOUNT =JCOUNT+1
C
C9        DEFINE MODE SEQUENCE AND SUBSYSTEMS USED
C
      DO 530 I=1,LCEI
  530 KU(I)=0
      LTEST=1
      DO 640 L = 1,LE
      LEQ=LEE(L)
      DO 540 K= 1,LEQ
  540 KS(L,K)=0
      IF(L.EQ.LQSVL.OR.LTEST.EQ.1) GO TO 550
      GO TO 640
  550 DO 630 K=1,LEQ
      IF(K.GE.LLQS.OR.LTEST.EQ.1) GO TO 560
      GO TO 630
  560 IF(.NOT.MDT.AND.MD(L,K)) GO TO 580
      GO TO 590
  580 LQSVL=MQ(L,K)
      LLQS=MMQ(L,K)
      LTEST=0
      IF(LQSVL.EQ.L) GO TO 630
      GO TO 640
  590 LTEST=1
      DO 600 M=1,INCEI
      IF(KW(M).LT.LMA(L,K,M))     GO TO 620
  600 CONTINUE
      DO 610 M=1,INCEI
      IF(KW(M).GT.LMA(L,K,M)     .OR.KW(M).EQ.0) GO TO 610
      IF(KW(M).EQ.LMA(L,K,M)     .AND.KW(M).EQ.1) KU(M)=1
  610 CONTINUE
      KS(L,K)=1
      LQSVL=LQ(L,K)
      LLQS=LLQ(L,K)
      LTEST=0
  620 IF(LQSVL.GT.L)GO TO 640
  630 CONTINUE
  640 CONTINUE
      J=JCAP+1
```

```
C
C10       ASSIGN MODE SEQUENCE NUMBERS
C
  650  J=J-1
       IF(J.NE.0) GO TO 680
       JCAP=JCAP+1
       J=JCAP
       DO 660 L = 1,LE
       LEQ=LEE(L)
       DO 660 K=1,LEQ
       JES(L,K,J)=KS(L,K)
  660  CONTINUE
       DO 670 IU=1,INCEI
       JU(IU,J)=KU(IU)
  670  CONTINUE
       GO TO 700
  680  DO 690 L = 1,LE
       LEQ=LEE(L)
       DO 690 K=1,LEQ
       IF(JES(L,K,J).NE.KS(L,K)) GO TO 650
  690  CONTINUE
  700  CONTINUE
       IJN(JCOUNT)=J
C
C11      REPOSITION ZEROS IN STATE BINARY VECTOR (KW)
C
  720  IF(NZ.LT.1.OR.NZ.GT.8)GO TO 790
       GO TO (770,760,750,740,730,728,726,724), NZ
  724  KW(LL8)=1
       LL8=LL8+1
       IF((LL8-8).LE.(INCEI-NZ)) GO TO 393
  726  KW(LL7)=1
       LL7=LL7+1
       IF((LL7-7).LE.(INCEI-NZ)) GO TO 392
  728  KW(LL6)=1
       LL6=LL6+1
       IF((LL6-6).LE.(INCEI-NZ)) GO TO 391
  730  KW(LL5)=1
       LL5=LL5+1
       IF((LL5-5).LE.(INCEI-NZ))GO TO 390
  740  KW(LL4)=1
       LL4=LL4+1
       IF((LL4-4).LE.(INCEI-NZ))GO TO 380
  750  KW(LL3)=1
       LL3=LL3+1
       IF((LL3-3).LE.(INCEI-NZ))GO TO 370
  760  KW(LL2)=1
       LL2=LL2+1
       IF((LL2-2).LE.(INCEI-NZ))GO TO 360
  770  KW(LL1)=1
       LL1=LL1+1
       IF((LL1-1).LE.(INCEI-NZ))GO TO 350
  790  CONTINUE
       NZ=NZ+1
       IF(NZ.LE.NZT)GO TO 330
       NZ=0
```

```
C
C12      WRITE MODE SEQUENCE AND SUBSYSTEMS USED
C
       DO 820 J=1,JCAP
      WRITE(6,1170)
      WRITE(6,1140) J
      DO 800 L = 1,LE
      LEQ=LEE(L)
      DO 800 K=1,LEQ
      IF(JES(L,K,J).EQ.0) GO TO 800
      WRITE(6,1150) (FNAME(L,K,I   ),I=1,10)
  800 CONTINUE
      IF(NC.NE.7) WRITE(6,1180)
      IF(NC.EQ.7) WRITE(6,1185)
      DO 810 M=1,INCEI
      IF(JU(M,J).NE.1) GO TO 810
      WRITE(6,1190) (CNAME(K,M),K=1,LCN8)
  810 CONTINUE
  820 CONTINUE
C
C13      WRITE MODE SEQUENCE ON TAPE FOR USE IN PROGRAM 2
C
      WRITE(4,1280) (IJN(IJK),IJK=1,JCOUNT)
      GO TO 70
  840 END FILE 4
      REWIND 4
      STOP
C
C14      DEFINE FORMATS
C
  950 FORMAT(10I3)
  960 FORMAT (26H1      SYSTEM CONFIGURATION)
  970 FORMAT(1H0,8X,9HEQUIPMENT,25X, 4HMTBF,7X,7HTHDM(1),6X,7HTHDM(2),
     *7X,7HTHDM(3),8X,7HTHDM(4)/56X,3HG D,11X,1HN,12X,1HB,15X,1HT//1X)
  980 FORMAT(I3,1X,8A4,F10.2/4E12.5,T65,A2)
  990 FORMAT(1H ,1X,I3,2X,8A4,F10.2,2X,4(E12.5,2X),3X,A2)
 1000 FORMAT (50H0  TOO MANY INPUTS OR M IS OUTSIDE ALLOWABLE RANGE)
 1010 FORMAT(I3,1X,F6.2,1X,I3,1X,7A4,2X,L1)
 1020 FORMAT(1H0,5X,9HEVENT NO.,I2,1X,3HIS ,3A4/5X,15HEVENT OCCURRED ,
     *F6.2,
     *20H HOURS AFTER TAKEOFF/5X,21HEVENT DESCRIPTION IS ,7A4/5X,L1)
 1060 FORMAT (1H0,7X,16HSUBFUNCTION/MODE,32X,21HEQUIPMENT DESCRIPTION,12
     *         X,18HMISSION DESCRIPTOR/1X)
 1065 FORMAT(1H0, 7X,16HSUBFUNCTION/MODE,32X,35HFLIGHT AND DETECTION TIM
     *E REMAINING)
 1070 FORMAT(4I3,1X,10A4,1X,2I3,1X,L1/23I3)
 1080 FORMAT(1X,2I3,2X,2I3,2X,10A4,T86,2I3,12X,L1,T59,23I1)
 1085 FORMAT(1X,2I3,2X,2I3,2X,10A4,                        T59,23I1)
 1090 FORMAT (29H0 TOO MANY INPUTS FOR F ARRAY)
 1140 FORMAT(1H0,11X,I3)
 1150 FORMAT(1H ,45X,10A4)
 1170 FORMAT(1H ,5X,16HMODE SEQUENCE NO,16X,18HSUBFUNCTIONAL FLOW)
 1180 FORMAT(1H0,58X,15HSUBSYSTEMS USED/1X)
 1185 FORMAT(1H0,58X,31HACTUAL AND APPARENT FLIGHT TIME)
 1190 FORMAT(1H ,60X,8A4)
 1270 FORMAT(23H1      EVENT DESCRIPTION)
 1280 FORMAT(25I3)
      END
```

```
C ***************************************************************************
C
C        MISSION DAMAGE EFFECTIVENESS MODEL - PROGRAM TWO
C                          MISDEM - PGM 2
C
C ***************************************************************************
      DIMENSION CEP(50),CN(3),CURVE(4,30),DEF(3),DNAME(8),FA(50),
     1   FTDNN(16,16),FTFNN(16),GRAPH(30),HDMT(4),ISUB(4),IT(23),ITT(23)
     2  ,IV(23),IZ(23),JZ(23),KW(23),KWW(23),MI(23) ,MTBF(23),OFF(3),
     3   PCAP(50),PCKILL(23,10,27),PCSURV(23,10,27),PI(256),PJ(256),
     4   PK(50),PM(23),PMM(23),PMISS(27,256),PTDN(16,16),PTFN(16),
     5   QO(256),QPRM(23,256),R(10),RA(23),RESUL(2),RM(24),THDM(23,4),
     6  TIMEF(23),TIMEN(23),WEAPN(7)
      INTEGER*2 IJN(256)
      LOGICAL MDT
      DATA OFF/4HOFFE,4HNSIV,4HE   /,DEF/4HDEFE,4HNSIV,4HE   /,OUT/2HNO/
C
C1       INITIALIZE
C
      LCN8=8
      LCEI=23
      L72=40
      INCEI=0
      DO 30 I=1,LCEI
      MTBF(I)=0.0
      PM(I)=0.0
      MI(I)=1
      IV(I)=I
      IZ(I)=I
      DO 20 MPRM=1,4
      THDM(I,MPRM)=0
   20 CONTINUE
   30 CONTINUE
      READ (5,1110) NZT,MCR,MPR,MAV,MLTH,NABORT
C
C2       READ AND WRITE SYSTEM DESCRIPTION
C
      WRITE (6,1120)
      WRITE (6,1130)
      DO 50 I=1,LCEI
      READ (5,1140) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),TMN
     1,TMF,ONOFF
      IF (M.EQ.999) GO TO 70
      WRITE (6,1150) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4)
      IF (M.GT.LCEI.OR.M.LT.1) GO TO 60
      IF (M.GT.INCEI) INCEI=I
      IF (ONOFF.EQ.OUT) MI(I)=0
      DO 40 MPRM=1,4
      THDM(I,MPRM)=HDMT(MPRM)
   40 CONTINUE
      MTBF(I)=TBFM
      TIMEN(I)=TMN
      TIMEF(I)=TMF
   50 CONTINUE
      READ (5,1140) M
      IF (M.EQ.999) GO TO 70
   60 WRITE (6,1160)
      STOP
   70 CONTINUE
```

```
C
C3      INITIALIZE
C
      ET=0.0
      II=0
      T1=0.0
      REWIND 4
      REWIND 3
C
C4      INITIALIZE
C
   80 CONTINUE
      DO 85 M=1,INCEI
      RM(M)=0.0
   85 CONTINUE
C
C5      READ AND WRITE EVENT DESCRIPTION
C
      READ (5,1170,END=1100) IEVENT,T2,NC,(WEAPN(I),I=1,7),MDT
      WRITE (6,1230)
      DO 90 N=1,3
      CN(N)=OFF(N)
      IF (NC.NE.0) CN(N)=DEF(N)
   90 CONTINUE
      WRITE (6,1180) IEVENT,(CN(N),N=1,3),T2,(WEAPN(I),I=1,7),MDT
      DO 100 L=1,INCEI
      IZ(L)=L
  100 CONTINUE
      IF (NC.EQ.0.OR.NC.EQ.6 .OR. NC.EQ.7) GO TO 200
C
C6      COMPUTE LETHAL RADII AND ASSIGN INDEX
C
      DO 120 I=1,NC
      READ (5,1240) NPOINT
      N2=2*NPOINT
      READ (5,1220) ISUB(I),(CURVE(I,J),J=1,N2)
      IS=ISUB(I)
      DO 120 M=1,INCEI
      DO 110 IJ=1,N2
      GRAPH(IJ)=CURVE(I,IJ)
  110 CONTINUE
      CALL TWODNZ (THDM(M,IS),1,GRAPH,NPOINT,2,RESUL)
      RA(M)=RESUL(2)
      RM(M)=AMAX1(RA(M),RM(M))
  120 CONTINUE
      INE=INCEI+1
      RM(INE)=0.0
      DO 140 LOO=1,INCEI
      INIT=LOO+1
      DO 130 J=INIT,INE
      IF (RM(J).LE.RM(LOO)) GO TO 130
      STORE=RM(J)
      RM(J)=RM(LOO)
      RM(LOO)=STORE
      ITORE=IZ(LOO)
      IZ(LOO)=IZ(J)
      IZ(J)=ITORE
  130 CONTINUE
  140 CONTINUE
      WRITE (6,1250) (RM(M),M=1,INE)
```

```
C
C7        COMPUTE SUBSYSTEM RELIABILITY IN TRANSITION
C
  200 CONTINUE
      DO 230 M=1,INCEI
      PM(M)=1.0
      IF (MI(M).EQ.0) GO TO 230
      IF (T2.LE.TIMEN(M)) GO TO 230
      IF (T2.GE.TIMEF(M)) GO TO 210
      DELTAT=T2-T1
      IF (T1.LE.TIMEN(M)) DELTAT=T2-TIMEN(M)
      GO TO 220
  210 DELTAT=0.0
      IF (TIMEF(M).GT.T1) DELTAT=TIMEF(M)-T1
  220 PM(M)=1.0-DELTAT/MTBF(M)+DELTAT**2/(2.0*MTBF(M)**2)
  230 CONTINUE
      T1=T2
      DO 240 L=1,INCEI
      JV=IV(L)
      PMM(L)=PM(JV)
  240 CONTINUE
C
C8       INITIALIZE STATE AND STATE PROBABILITIES
C
      NZ=0
      NZI=0
      DO 250 I=1,LCEI
      KW(I)=0
  250 CONTINUE
      DO 260 J=1,L72
      PCAP(J)=0.0
  260 CONTINUE
      JCAP=0
      IF (IEVENT.NE.1) GO TO 290
      ICOUNT=1
      DO 270 M=1,INCEI
      IT(M)=1
      ITT(M)=1
      IF (MI(M).EQ.1) GO TO 270
      IT(M)=0
      ITT(M)=0
  270 CONTINUE
C
C9        ABORT FLOW CONTROL
C
      IF(NABORT.NE.0)GO TO 275
      GO TO 278
C
C10       INITIALIZE ABORT STATES
C
  275 IF (I EVENT.NE.1) GO TO 290
      READ(3)JEVENT,JCOUNT,(PJ(L),L=1,JCOUNT)
      IF(JEVENT.NE.NABORT)GO TO 275
      DO 277 I=1,JCOUNT
      PI(I)=PJ(I)
  277 PJ(I)=0.0
      GO TO 305
```

```
C
C11       INITIALIZE NORMAL STATES
C
  278 DO 280 L=1,256
      PI(L)=1.0
      PJ(L)=0.0
  280 CONTINUE
      GO TO 310
  290 DO 300 I=1,JCOUNT
      PI(I)=PJ(I)
      PJ(I)=0.0
  300 CONTINUE
  305 CONTINUE
      ICOUNT=0
C
C12       GENERATE A PRIOR STATE OF THE SYSTEM (IT)
C
  310 IF(NABORT.EQ.0)GO TO 315
      GO TO 320
315   CONTINUE
      IF(IEVENT.EQ.1)GO TO 440
  320 CONTINUE
      DO 330 I=1,INCEI
      IT(I)=1
  330 CONTINUE
      IF (NZ.LT.1) GO TO 410
      II1=1
  360 IT(II1)=0
      IF (NZ.LT.2) GO TO 410
      II2=II1+1
  370 IT(II2)=0
      IF (NZ.LT.3) GO TO 410
      II3=II2+1
  380 IT(II3)=0
      IF (NZ.LT.4) GO TO 410
      II4=II3+1
  390 IT(II4)=0
      IF (NZ.LT.5) GO TO 410
      II5=II4+1
  400 IT(II5)=0
      IF(NZ.LT.6)GO TO 410
      II6=II5+1
  405 IT(II6)=0
      IF(NZ.LT.7)GO TO 410
      II7=II6+1
  406 IT(II7)=0
      IF(NZ.LT.8)GO TO 410
      II8=II7+1
  407 IT(II8)=0
```

```
C
C13      TEST FOR DELETED OR CRITICAL SUBSYSTEMS
C
  410 CONTINUE
      DO 420 MM=1,INCEI
      IF (MI(MM).EQ.1) GO TO 420
      IF (IT(MM).EQ.1) GO TO 840
  420 CONTINUE
      IF(NC.EQ.7) GO TO 425
      IF(MCR.EQ.0)GO TO 422
      IF(IT(MCR).EQ.0) GO TO 840
  422 IF(MPR.EQ.0)GO TO 424
      IF(IT(MPR).EQ.0) GO TO 840
  424 IF(MAV.EQ.0)GO TO 425
      IF(IT(MAV).EQ.0) GO TO 840
  425 ICOUNT=ICOUNT+1
C
C14      RESHUFFLE SUBSYSTEM ORDER
C
      DO 430 L=1,INCEI
      JV=IV(L)
      ITT(L)=IT(JV)
  430 CONTINUE
  440 JCOUNT=0
C
C15      GENERATE A CURRENT STATE OF THE SYSTEM (KW)
C
  450 CONTINUE
      DO 460 I=1,INCEI
      KW(I)=1
  460 CONTINUE
      IF (NZI.LT.1) GO TO 540
      LL1=1
  490 KW(LL1)=0
      IF (NZI.LT.2) GO TO 540
      LL2=LL1+1
  500 KW(LL2)=0
      IF (NZI.LT.3) GO TO 540
      LL3=LL2+1
  510 KW(LL3)=0
      IF (NZI.LT.4) GO TO 540
      LL4=LL3+1
  520 KW(LL4)=0
      IF (NZI.LT.5) GO TO 540
      LL5=LL4+1
  530 KW(LL5)=0
      IF(NZI.LT.6)GO TO 540
      LL6=LL5+1
  535 KW(LL6)=0
      IF(NZI.LT.7)GO TO 540
      LL7=LL6+1
  536 KW(LL7)=0
      IF(NZI.LT.8)GO TO 540
      LL8=LL7+1
  537 KW(LL8)=0
  540 CONTINUE
```

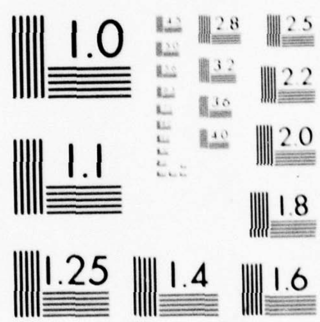
AD-A070 991 ROCKWELL INTERNATIONAL LOS ANGELES CALIF LOS ANGELE--ETC F/G 1/3
MISDEM COMPUTER SIMULATION. VOLUME II. ANALYST MANUAL.(U)
MAY 79 G L GALLIEN, S C SILVER N00123-76-C-0159

UNCLASSIFIED NA-76-372 JTCG/AS-76-S-004 NL

2 OF 2

AD
A070991

END
DATE
FILMED
8-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
C
C16        TEST FOR SUPPRESSED SUBSYSTEMS
C
      DO 550 MM=1,INCEI
      IF (MI(MM).EQ.1) GO TO 550
      IF (KW(MM).EQ.1) GO TO 710
  550 CONTINUE
      IF(NC.EQ.7)GO TO 555
      IF(MCR.EQ.0)GO TO 552
      IF(KW(MCR).EQ.0)GO TO 710
  552 IF(MPR.EQ.0)GO TO 554
      IF(KW(MPR).EQ.0)GO TO 710
  554 IF(MAV.EQ.0)GO TO 555
      IF(KW(MAV).EQ.0)GO TO 710
  555 JCOUNT=JCOUNT+1
C
C17        RESHUFFLE SUBSYSTEM ORDER
C
      DO 560 L=1,INCEI
      JV=IV(L)
      KWW(L)=KW(JV)
  560 CONTINUE
C
C18       COMPUTE THE SYSTEM STATE TRANSITION PROBABILITY
C
      IF (II.NE.6) GO TO 1500
C
C19       QUICK CONVENTIONAL THREAT DAMAGE AND RELIABILITY
C
      SUMTRK=0.0
      DO 620 K=1,KMAX
      SUMTRA=0.0
      DO 610 L=1,LMAX
      TRANS=1.0
      DO 600 M=1,INCEI
      PCSURV(M,L,K) = 1.0 - PCKILL(M,L,K)
      IF (ITT(M)-KWW(M)) 570,580,590
  570 TRANS=0.0
      GO TO 620
  580 IF (ITT(M).EQ.0.0) GO TO 600
      TRANS=PCSURV(M,L,K)*PMM(M)*TRANS
      GO TO 600
  590 TRANS=(1.0-PCSURV(M,L,K)*PMM(M))*TRANS
  600 CONTINUE
      SUMTRA=TRANS+SUMTRA
  610 CONTINUE
      SUMTRK=(SUMTRA/LMAX)*PMISS(K,ICOUNT)+SUMTRK
  620 CONTINUE
      TRANS=SUMTRK
      GO TO 700
 1500 IF(II.NE.7) GO TO 630
```

```
C
C20     SLOW THREAT DAMAGE
C
        TRANS = 1.0
        NTFN=0
        ITFN=0
        IINCEI=INCEI/2
        DO 1510 L=1,IINCEI
        IF(KWW(L).EQ.1)NTFN=2**((INCEI/2)-L)+NTFN
  1510  IF(ITT(L).EQ.1)ITFN=2**((INCEI/2)-L)+ITFN
        NTDN=0
        ITDN=0
        JINCEI=IINCEI+1
        DO 1520 L=JINCEI,INCEI
        IF(KWW(L).EQ.1)NTDN=2**(INCEI-L)+NTDN
  1520  IF(ITT(L).EQ.1)ITDN=2**(INCEI-L)+ITDN
        NTTF=NTFN+1
        ITTF=ITFN+1
        NTTD=NTDN+1
        ITTD=ITDN+1
        IF(ITTD.GT.ITTF.OR.NTTD.GT.NTTF)GO TO 1605
        IF(ITTF.LE.1.OR.ITTD.LE.1) GO TO 1605
        IF(NTTF.GT.(ITTF-1).OR.NTTD.GT.(ITTD-1))GO TO 1605
        GO TO 1610
  1605  TRANS1=0
        GO TO 1720
  1610  TRANS1=0
        IF(NTTF-(ITTF-1))1670,1620,1620
  1620  IF(NTTD-(ITTD-1))1630,1625,1625
  1625  DO 1627 I=NTTF,MLTH
        SUMPTD=0
        DO 1626 J=NTTD,I
  1626  SUMPTD=PTDN(I,J)+SUMPTD
  1627  TRANS1=PTFN(I)*SUMPTD+TRANS1
        GO TO 1720
  1630  DO 1631 I=NTTF,MLTH
  1631  TRANS1=PTFN(I)*PTDN(I,NTTD)+TRANS1
        GO TO 1720
  1670  IF(NTTD-(ITTD-1))1680,1675,1675
  1675  DO 1676 J=NTTD,NTTF
  1676  TRANS1=PTDN(NTTF,J)+TRANS1
        TRANS1=PTFN(NTTF)*TRANS1
        GO TO 1720
  1680  TRANS1=PTFN(NTTF)*PTDN(NTTF,NTTD)
  1720  TRANS=TRANS1*TRANS
        GO TO 700
```

```
C
C21      QUICK NUCLEAR DAMAGE AND RELIABILITY
C         (ALSO NON-DEFENSIVE TRANSITIONS)
C
  630 TRANS=1.0
      NF=1
      IF(II.EQ.0) GO TO 640
      TRANS=QQ(ICOUNT)
      NF=0
  640 CONTINUE
      DO 690 M=1,INCEI
      IF (ITT(M)-KWW(M)) 650,660,670
  650 TRANS=0.0
      GO TO 700
  660 IF (ITT(M).EQ.0) GO TO 680
      NF=1
      TRANS=TRANS*PMM(M)
      GO TO 690
  670 TRANS=TRANS*(1.0-PMM(M))
  680 IF (NF.EQ.0) TRANS=TRANS+QPRM(M,ICOUNT)
  690 CONTINUE
C
C22          COMPUTE THE STATE PROBABILITIES
C
  700 PJ(JCOUNT)=TRANS*PI(ICOUNT)+PJ(JCOUNT)
C
C23          RESET CURRENT STATE (KW) ZERO LOCATIONS
C
  710 IF (NZI.LT.1.OR.NZI.GT.8) GO TO 820
      GO TO (800,780,760,740,720,715,714,713),NZI
  713 KW(LL8)=1
      LL8=LL8+1
      IF((LL8-8).LE.(INCEI-NZI))GO TO 537
  714 KW(LL7)=1
      LL7=LL7+1
      IF((LL7-7).LE.(INCEI-NZI))GO TO 536
  715 KW(LL6)=1
      LL6=LL6+1
      IF((LL6-6).LE.(INCEI-NZI))GO TO 535
  720 KW(LL5)=1
      LL5=LL5+1
      IF((LL5-5).LE.(INCEI-NZI))GO TO 530
  740 KW(LL4)=1
      LL4=LL4+1
      IF((LL4-4).LE.(INCEI-NZI))GO TO 520
  760 KW(LL3)=1
      LL3=LL3+1
      IF((LL3-3).LE.(INCEI-NZI))GO TO 510
  780 KW(LL2)=1
      LL2=LL2+1
      IF((LL2-2).LE.(INCEI-NZI))GO TO 500
  800 KW(LL1)=1
      LL1=LL1+1
      IF((LL1-1).LE.(INCEI-NZI))GO TO 490
  820 CONTINUE
      NZI=NZI+1
      IF (NZI.LE.NZT) GO TO 450
      NZI=0
```

```
C
C24       ABORT FLOW CONTROL
C
      IF(NABORT.EQ.0)GO TO 835
      GO TO 840
  835 CONTINUE
      IF (IEVENT.EQ.1) GO TO 970
C
C25       RESET PRIOR STATE (IT) ZERO LOCATIONS
C
  840 IF (NZ.LT.1.OR.NZ.GT.8) GO TO 950
      GO TO (930,910,890,870,850,845,844,843), NZ
  843 IT(II8)=1
      II8=II8+1
      IF((II8-8).LE.(INCEI-NZ))GO TO 407
  844 IT(II7)=1
      II7=II7+1
      IF((II7-7).LE.(INCEI-NZ))GO TO 406
  845 IT(II6)=1
      II6=II6+1
      IF((II6-6).LE.(INCEI-NZ))GO TO 405
  850 IT(II5)=1
      II5=II5+1
      IF((II5-5).LE.(INCEI-NZ))GO TO 400
  870 IT(II4)=1
      II4=II4+1
      IF((II4-4).LE.(INCEI-NZ)) GO TO 390
  890 IT(II3)=1
      II3=II3+1
      IF((II3-3).LE.(INCEI-NZ)) GO TO 380
  910 IT(II2)=1
      II2=II2+1
      IF((II2-2).LE.(INCEI-NZ)) GO TO 370
  930 IT(II1)=1
      II1=II1+1
      IF((II1-1).LE.(INCEI-NZ))GO TO 360
  950 CONTINUE
      NZ=NZ+1
      IF (NZ.LE.NZT) GO TO 320
      NZ=0
  970 CONTINUE
C
C26      WRITE OUTPUT STATE PROBABILITY TAPE
C
      IF(NABORT.NE.0)GO TO 979
      JEVENT=IEVENT
      WRITE(3) JEVENT,JCOUNT,(PJ(L),L=1,JCOUNT)
C
C27     COMPUTE THE MODE SEQUENCE PROBABILITIES
C
  979 READ (4,1280) (IJN(IJK),IJK=1,JCOUNT)
      JCAP=0
      DO 980 IJK=1,JCOUNT
      J=IJN(IJK)
      PCAP(J)=PCAP(J)+PJ(IJK)
      JCAP=MAX0(JCAP,J)
  980 CONTINUE
      PARIVE=PCAP(1)+PCAP(2)+PCAP(3)
```

```
C
C28      READ AND WRITE OUTPUT MODE PROBABILITIES AND CAPABILITES
C
      DO 1000 J=1,JCAP
      READ (5,1260) CEP(J),FA(J),PK(J)
      IF (NC.EQ.0) GO TO 990
      IF(NC.NE.7) WRITE (6,1270) J,PCAP(J),CEP(J),FA(J)
      IF(NC.EQ.7) WRITE(6,1275) J,PCAP(J)
      GO TO 1000
  990 WRITE (6,1190) J,PCAP(J),PK(J)
 1000 CONTINUE
      IF(NC.NE.7) GO TO 9030
      WRITE(6,9020)PARIVE
 9020 FORMAT(1X,'PARIVE = ',E12.5)
C
C29      IF EVENT IS OFFENSIVE,COMPUTE ET - MISSION EFFECTIVENESS
C
 9030 II=0
      IF (NC.NE.0) GO TO 1030
      DO 1010 J=1,INCEI
      IV(J)=J
 1010 CONTINUE
      SUM=0.0
      DO 1020 J=1,JCAP
      SUM=SUM+PCAP(J)*PK(J)
 1020 CONTINUE
      WRITE (6,1200) IEVENT,SUM
      ET=ET+SUM
      WRITE (6,1210) ET
      GO TO 80
C
C
C30      IF EVENT IS DEFENSIVE DETERMINE SYSTEM SURVIVABILITY PARAMETERS
C
C
 1030 IF(NC.EQ.7) GO TO 2500
      II=1
      DO 1040 IQ=1,INCEI
      IV(IQ)=IZ(IQ)
 1040 CONTINUE
      IF (NC.NE.6) GO TO 1070
      II=6
```

```
C
C31      QUICK CONVENTIONAL KILLS
C
      READ (5,1290) LMAX,KMAX,(R(K),K=1,KMAX)
      WRITE (6,1300) (R(K),K=1,KMAX)
      DO 1050 M=1,INCEI
      DO 1050 L=1,LMAX
      READ (5,1310) (PCKILL(M,L,K),K=1,KMAX)
 1050 WRITE (6,1320) M,L,(PCKILL(M,L,K),K=1,KMAX)
      KMAX1=KMAX-1
      DO 1060 IJK=1,JCOUNT
      PMISS(KMAX,IJK)=1.0
      J=IJN(IJK)
      SIGMA=CEP(J)/1.178
      DENUM=2.0*SIGMA**2
      DEN175=175.0*DENUM
      EX1=1.0
      DO 1060 K=1,KMAX1
      EX2=0.0
      RM2=R(K)**2
      IF (RM2.LT.DEN175) EX2=EXP(-RM2/DENUM)
      PMISS(K,IJK)=EX1-EX2
      PMISS(KMAX,IJK)=PMISS(KMAX,IJK)-PMISS(K,IJK)
      EX1=EX2
 1060 CONTINUE
      GO TO 80
C
C32      SLOW KILLS
C
 2500 II=7
      DO 2122 I=1,MLTH
      READ(5,2120) FTFNN(I),(FTDNN(I,J),J=1,MLTH)
 2120 FORMAT(17F3.2)
      IFF=I-1
      IF(I.EQ.1)   GO TO 2125
      PTFN(I) = (FTFNN(I)-FTFNN(IFF))
      GO TO 2129
 2125 PTFN(I)=FTFNN(I)
 2129 CONTINUE
       DO 2130 J=1,MLTH
      JFF=J-1
      IF(J.EQ.1)GO TO 2140
      PTDN(I,J)=FTDNN(I,J)-FTDNN(I,JFF)
      GO TO 2130
 2140 PTDN(I,J) = FTDNN(I,J)
 2130 CONTINUE
 2122 CONTINUE
      GO TO 80
```

```
C
C 33      QUICK NUCLEAR KILLS
C
 1070 CONTINUE
      DO 1090 IJK=1,JCOUNT
      QO(IJK)=0.0
      J=IJN(IJK)
      SIGMA=CEP(J)/1.178
      DENUM=2.0*SIGMA**2
      DEN175=175.0*DENUM
      EX1=0.0
      RM2=RM(1)**2
      IF (RM2.LT.DEN175) EX1=EXP(-RM2/DENUM)
      QO(IJK)=EX1**FA(J)
      DO 1080 M=1,INCEI
      QPRM(M,IJK)=0.0
      RMM2=RM(M)**2
      RMM12=RM(M+1)**2
      IF (RMM2.LE.RMM12.OR.RMM12.GE.DEN175) GO TO 1080
      EXM1=EXP(-RMM12/DENUM)
      EXM=0.0
      IF (RMM2.LT.DEN175) EXM=EXP(-RMM2/DENUM)
      QPRM(M,IJK)=EXM1**FA(J)-EXM**FA(J)
 1080 CONTINUE
 1090 CONTINUE
      GO TO 80
 1100 REWIND 4
      REWIND 3
      STOP
C
C 34      DEFINE FORMATS
C
 1110 FORMAT (10I3)
 1120 FORMAT (26H1        SYSTEM CONFIGURATION)
 1130 FORMAT (1H0,8X,9HEQUIPMENT,25X,4HMTBF,7X,7HTHDOM(1),6X,7HTHDOM(2),7X
     1,7HTHDOM(3),8X,7HTHDOM(4)/56X,3HG D,11X,1HN,12X,1HB,15X,1HT//1X)
 1140 FORMAT (I3,1X,8A4,F10.2/4E12.5,2X,2F6.2,2X,A2)
 1150 FORMAT (1H ,1X,I3,2X,8A4,F10.2,2X,4(E12.5,2X))
 1160 FORMAT (50H0  TOO MANY INPUTS OR M IS OUTSIDE ALLOWABLE RANGE)
 1170 FORMAT (I3,1X,F6.2,1X,I3,1X,7A4,2X,L1)
 1180 FORMAT (1H0,5X,9HEVENT NO.,I2,1X,3HIS ,3A4/5X,15HEVENT OCCURRED ,F
     16.2,20H HOURS AFTER TAKEOFF/5X,21HEVENT DESCRIPTION IS ,7A4 /5X,L1
     2)
 1190 FORMAT (1H0,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5,2X,7HPK(J) =,E12.5)
 1200 FORMAT (1H0,10X,40HEFFECTIVENESS FOR OFFENSIVE EVENT NUMBER,I3,1X,
     12HIS,E12.5)
 1210 FORMAT (1H0,10X,36HCUMULATIVE MISSION EFFECTIVENESS IS ,E12.5)
 1220 FORMAT (I3,2X,6E12.5/4(5X,6E12.5))
 1230 FORMAT (23H1        EVENT DESCRIPTION)
 1240 FORMAT (I3)
 1250 FORMAT (1H0 //5X,8HRM ARRAY/(14X,4(E12.5,2X)))
 1260 FORMAT (3E12.5)
 1270 FORMAT (1H0,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5,2X,8HCEP(J) =,E12.5,
     12X,7HFA(J) =,E12.5)
```

```
1275 FORMAT( 1H0,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5)
1280 FORMAT (25I3)
1290 FORMAT (2I3,10F7.0)
1300 FORMAT (1H0,2X,31HCOMPONENT PROBABILITIES OF KILL/
   1 17H  COMP# ELEV#  R=,10(F8.0,1X))
1310 FORMAT (8E10.4)
1320 FORMAT (2I6,2(4X,E10.4))
     END

     SUBROUTINE TWODNZ (XG,NIZ,Z1,NX,NZ,ANS1)
C
C1       TWO DIMENSIONAL LINEAR INTERPOLATION ROUTINE
C
     DIMENSION Z1(1), ANS1(1)
     IXL=(NIZ-1)*NX+1
C2       IF GIVEN VALUE IS LESS THAN LOWER LIMITS OF TABLE, SET RESULTS
C                 EQUAL TO LOWER LIMIT
     IF (XG-Z1(IXL)) 70,70,10
  10 LL=IXL+1
     LU=IXL+NX-1
C3       SEARCH FOR INTERVAL IN WHICH GIVEN VALUE LIES
     DO 20 J=LL,LU
     IF (XG-Z1(J)) 30,80,20
  20 CONTINUE
C4       IF GIVEN VALUE IS GREATER THAN UPPER LIMIT OF TABLE, SET
C                 RESULTS EQUAL TO UPPER LIMIT
     J=IXL+NX-1
     GO TO 80
C5       GIVEN VALUE WITHIN TABLE, CALCULATE INTERPOLATION FACTOR
  30 RAT=(Z1(J)-XG)/(Z1(J)-Z1(J-1))
  40 JP=J-(NIZ-1)*NX
C6       DO LINEAR INTERPOLATION FOR RESULTS
     DO 60 K=1,NZ
C7       CHECK FOR ZERO SUBSCRIPT
     IF (JP.EQ.1) GO TO 50
     ANS1(K)=Z1(JP)-RAT*(Z1(JP)-Z1(JP-1))
     GO TO 60
  50 ANS1(1)=Z1(1)
  60 JP=JP+NX
     RETURN
  70 J=IXL
  80 RAT=0.0
     GO TO 40
     END
```

ELECTRONICS MODE TEST DECK

```
   2  0  0  0
   1 ELECTRONICS A                                  10.
           20.          10.
   2 ELECTRONICS B                                  20.
           10.          20.
999
   1       .10  6 23MM QUAD, POSITION 4         T
   1   1   1   2 MILITARY FUNCTION A                       0   0 F

   1   2   2   1   NORMAL MODE                             0   0 F
   1
   1   3   5   5   COMPLETE FAILURE                        0   0 F

   0 99    0   0

   2   1   2   2 MILITARY FUNCTION B                       0   0 F

   2   2   5   5   NORMAL MODE                             0   0 F
   2
   2   3   5   5   DEGRADED MODE                           0   0 F

   0 99    0   0

999    0   0   0

   2       .20  6 23MM QUAD, POSITION 4         T
   1   1   1   2 MILITARY FUNCTION A                       0   0 F

   1   2   2   1   NORMAL MODE                             0   0 F
   1
   1   3   5   5   COMPLETE FAILURE                        0   0 F

   0 99    0   0

   2   1   2   2 MILITARY FUNCTION B                       0   0 F

   2   2   5   5   NORMAL MODE                             0   0 F
```

```
 2
 2   3   5   5    DEGRADED MODE                                   0   0  F

 0  99   0   0

999  0   0   0

 3      .30   6 23MM QUAD, POSITION 4          T
 1   1   1   2 MILITARY FUNCTION A                                0   0  F

 1   2   2   1    NORMAL MODE                                     0   0  F
 1
 1   3   5   5    COMPLETE FAILURE                                0   0  F

 0  99   0   0

 2   1   2   2 MILITARY FUNCTION B                                0   0  F

 2   2   5   5    NORMAL MODE                                     0   0  F
 2
 2   3   5   5    DEGRADED MODE                                   0   0  F

 0  99   0   0

999  0   0   0

 4      .30   0 MARK 82 SNAKEYE                F
 1   1   1   2 MILITARY FUNCTION A                                0   0  F

 1   2   2   1    NORMAL MODE                                     0   0  F
 1
 1   3   5   5    COMPLETE FAILURE                                0   0  F

 0  99   0   0

 2   1   2   2 MILITARY FUNCTION B                                0   0  F

 2   2   5   5    NORMAL MODE                                     0   0  F
```

```
  2
  2   3   5   5    DEGRADED MODE                              0   0 F

  0  99   0   0

999   0   0   0

  5       .40   6 23MM QUAD, POSITION 4         T
  1   1   1   2 MILITARY FUNCTION A                           0   0 F

  1   2   2   1    NORMAL MODE                                0   0 F
  1
  1   3   5   5    COMPLETE FAILURE                           0   0 F

  0  99   0   0

  2   1   2   2 MILITARY FUNCTION B                           0   0 F

  2   2   5   5    NORMAL MODE                                0   0 F
  2
  2   3   5   5    DEGRADED MODE                              5   5 T

  0  99   0   0

999   0   0   0

  6       .50   0 LAND AT BASE                  T
  1   1   1   2 MILITARY FUNCTION A                           0   0 F

  1   2   5   5    NORMAL MODE                                0   0 F

  0  99   0   0

999   0   0   0
```

```
  2  0  0  0  0  0
  1 ELECTRONICS A                                     10.
                                                              .1     .6
  2 ELECTRONICS B                                     20.
                                                              .1     .6
999
  1     .10    6 23MM QUAD,POSITION 4                  T
           10.          1.
           10.          1.
           10.          1.
  1   2    100.   1000.
     0.     0.
     .1     0.
  2     .20    6 23MM QUAD,POSITION 4                  T
           10.          1.
           10.          1.
           10.          1.
  1   2    100.   1000.
     0.     0.
     .1     0.
  3     .30    6 23MM QUAD,POSITION 4                  T
           10.          1.
           10.          1.
           10.          1.
  1   2    100.   1000.
     0.     0.
     .1     0.
  4     .30    0 MARK 82 SNAKEYE                       F
                                          0.9
                                          0.0
                                          0.6
  5     .40    6 23MM QUAD,POSITION 4                  T
           10.          1.
           10.          1.
           10.          1.
  1   2    100.   1000.
     0.     0.
     .1     0.
  6     .40    0 LAND AT BASE                          T
                                          0.0
```

VEHICLE MODE TEST DECK – PROGRAM 1

```
  4  1  1 1
  1 VEH ACTUALLY GOOD FOR 2 OR 3 DT      10000.

  2 VEH  ACTUALLY GOOD FOR DELTA T       10000.

  3 VEH APPARENTLY GOOD FOR 2OR3 DT      10000.

  4 VEH  APPARENTLY GOOD FOR DELTA T     10000.

999

  1     .10    7 23MM QUAD, POSITION 4          T
  1  1  1  2 FLIGHT FUNCTION                           0  0 F

  1  2  5  5   NORMAL MODE                             0  0 F

  0 99  0  0

999  0  0  0

  2     .20    7 23MM QUAD, POSITION 4          T
  1  1  1  2 FLIGHT FUNCTION                           0  0 F

  1  2  5  5   NORMAL MODE A                           0  0 F
  3
  1  3  5  5   NORMAL MODE B                           0  0 F
  4
  1  4  5  5   NORMAL MODE C                           0  0 F
  1  2
  1  5  5  5   ABORT MODE A                            0  0 F
  1
  1  6  5  5   ABORT MODE B                            0  0 F
  2
  1  7  5  5   DOWN                                    0  0 F

  0 99  0  0

999  0  0  0
```

```
  3     .30    7 23MM QUAD, POSITION 4          T
  1   1   1  2 FLIGHT FUNCTION                                  0  0 F

  1   2   5  5   NORMAL MODE A                                  0  0 F
  3
  1   3   5  5   NORMAL MODE B                                  0  0 F
  4
  1   4   5  5   NORMAL MODE C                                  0  0 F
  1
  1   5   5  5   ABORT MODE A                                   0  0 F
  2
  1   6   5  5   DOWN                                           0  0 F

  0  99   0  0

999   0   0  0

  5     .40    7 23MM QUAD, POSITION 4          T
  1   1   1  2 FLIGHT FUNCTION                                  0  0 F

  1   2   5  5   NORMAL MODE A                                  0  0 F
  4
  1   3   5  5   NORMAL MODE B                                  0  0 F
  1
  1   4   5  5   NORMAL MODE C                                  0  0 F
  2
  1   5   5  5   DOWN                                           0  0 F

  0  99   0  0

999   0   0  0

  6     .50    7 LAND AT BASE                   T
  1   1   1  2 FLIGHT FUNCTION                                  0  0 F

  1   2   5  5   NORMAL MODE A                                  0  0 F
  4

  1   3   5  5   NORMAL MODE B                                  0  0 F
  1
  1   4   5  5   NORMAL MODE C                                  0  0 F
  2
  1   5   5  5   DOWN                                           0  0 F

  0  99   0  0

999   0   0  0
```

VEHICLE MODE TEST DECK – PROGRAM 2

```
   4  1  1  1  4  0
   1 VEH ACTUALLY GOOD FOR 2 OR 3 DT         10000.
                                                          0.    1.
   2 VEH  ACTUALLY GOOD FOR DELTA T          10000.
                                                          0.    1.
   3 VEH APPARENTLY GOOD FOR 2OR3 DT         10000.
                                                          0.    1.
   4 VEH  APPARENTLY GOOD FOR DELTA T        10000.
                                                          0.    1.
 999

   1     .10    7 23MM QUAD, POSITION 4
         10.               1.
  02100100100100
  04 50100100100
  05 30 60 90100
 100 20 40 70100
   2     .20    7 23MM QUAD, POSITION 4
         10.               1.
         10.               1.
         10.               1.
         10.               1.
         10.               1.
         10.               1.
  02100100100100
  04 50100100100
  05 30 60 90100
 100 20 40 70100
   3     .30    7 23MM QUAD, POSITION 4
         10.               1.
         10.               1.
         10.               1.
         10.               1.
         10.               1.
  02100100100100
  04 50100100100
  05 30 60 90100
 100 20 40 70100
```

```
  5      .40     7 23MM QUAD, POSITION 4
        10.             1.
        10.             1.
        10.             1.
        10.             1.
 021001001001 00
 04 50100100100
 05 30 60 90100
100 20 40 70100
  6      .50     7 LAND AT BASE
        10.             1.
        10.             1.
        10.             1.
        10.             1.
 021001001001 00
 04 50100100100
 05 30 60 90100
100 20 40 70100
```

SIMULATION MODEL

This section discusses the manner in which calculations are performed within the computer routines. The blocks of code are headed by comment cards. These comment cards are also the titles of the schematic diagrams (Figures 21 through 49), which facilitate cross-referencing. Following the discussion of the code is a subsection entitled User Information, which describes restrictions, simulation errors, and limitations affecting input. This is followed by a description of abbreviations and symbols for the simulation model.

PROGRAM 1

Program 1 consists of only one routine. The purpose of this routine is to define and number all possible mode sequences for all events resulting from all possible states of the system. The mode sequence numbers are an array that is stored on an intermediate device for use in Program 2.

The statements

```
C ************************************************************************
C
C        MISSION DAMAGE EFFECTIVENESS MODEL - PROGRAM ONE
C                         MISDEM - PGM 1
C
C ************************************************************************
      DIMENSION CN(3),CNAME(8,23),DEF(3),DNAME(8),FNAME(10,27,10),
     1  HDMT(4),JES(10,27,25),JU(23,25),KS(10,27),KU(23),KW(23),LEE(10),
     2  LF(10),LLF(27),LLQ(10,27),LMA(10,27,23),LMAT(10,27,23),LMATT(23)
     3  ,LQ(10,27),MD(10,27),MI(23),MMQ(10,27),MQ(10,27),OFF(3),WEAPN(7)
      INTEGER*2 IJN(256)
      LOGICAL MD,MDT
      EQUIVALENCE (LMA(1,1,1),LMAT(1,1,1))
      DATA BLK/4H    /,OFF/4HOFFE,4HNSIV,4HE   /,DEF/4HDEFE,4HNSIV,
     *4HE   /,OUT/2HNO/
```

are used to allocate storage by use of the DIMENSION statement, to reserve two integer storage locations for each of the 256 INTEGER*2 words, to declare the variables MD and MDT as logical with four storage locations allocated for each, to define storage that is to be shared by two or more entities by use of the EQUIVALENCE statement, and to define the initial values for various variables by use of the DATA statement.

The statements

```
C
C 1      INITIALIZE
C
      L1=27
      LCN8=8
      L40=10
      LCEI=23
      L72=7
```

are used to define variable names for the dimensional constant. L1 is the maximum number of modes (in all subfunctions) of the mode logic (see Figure 21). L40 is the maximum number of subfunctions. LCN8 is the maximum number of equipment name segments. L72 is the maximum number of mode sequences (or flow paths). LCEI is the maximum number of subsystems in the equipment configuration, used as a check (later in the program) on program input.

The statements

```
      INCEI=0
      DO 20 I=1,LCEI
      MI(I)=1
      DO 20 J=1,LCN8
      CNAME(J,I)=BLK
   20 CONTINUE
```

are used to initialize subsystems counter (INCEI) to zero, to set the on-off flag to "on" and to insert blanks in all the subsystem name locations.

The statement

```
      READ(5,950)NZT,MCR,MPR,MAV
```

is used to read the first input data card. These variables are used for control of state generation (through NZT) and state probability generation (through MCR, MPR and MAV). These four inputs are discussed at length in the User Information subsection.

The statements

```
C
C?        READ AND WRITE SYSTEM DESCRIPTION
C
      WRITE(6,960)
      WRITE(6,970)
      DO 40 I=1,LCEI
      READ(5,980) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),ONOFF
      IF(M.EQ.999)GO TO 60
      WRITE(6,990)M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),ONOFF
      IF(M.GT.LCEI.OR.M.LT.1) GO TO 50
      IF(M.GT.INCEI)INCEI=M
      IF(ONOFF.EQ.OUT)MI(I)=0
      DO 40 J=1,LCN8
      CNAME(J,I)=DNAME(J)
   40 CONTINUE
      READ(5,980) M
      IF(M.EQ.999) GO TO 60
   50 WRITE(6,1000)
      STOP
   60 CONTINUE
```

are used for input/output of the system description. The first two lines write the headings in preparation for the output. The next statement starts a loop which allows for the maximum input of 23 subsystem definitions. Each subsystem definition consists of the equipment name, mean time between failures, nuclear damage thresholds and a logical flag (ONOFF) which can be used to suppress a subsystem (prevent its being considered further). If the system number that is input is equal to 999, the program switches control out of the loop (i.e., "60 CONTINUE") signifying that there are no more inputs of this type. However, if a system configuration is input, the program proceeds to write out information. The next statement is used to check for blanks (subsystems left out) or too many subsystems (relative to LCEI) in the input data. If either one of these conditions exist, program control is transferred to an error message which is written out and the program stops. The next statement updates the counter which keeps track of the total number of subsystems input. The next statement is used to set a flag if a subsystem is being suppressed. The next three statements store the subsystem name by a reference index number corresponding to LCEI. Program control is then transferred to the beginning of the system configuration loop to read the next subsystem description. Should the program go through the loop 23 times and never transfer control to the statement which says "60 CONTINUE", the program will read another card. If this card contains another system description, the program will write an error message and stop; however, if the input contains a "999" for the subsystem number, control will be transferred to the "60 CONTINUE" statement and program execution will flow as described in the next set of statements.

The statement

```
      REWIND 4
```

is used to rewind the tape to the beginning on which the mode sequence number array is to be written.

The statement

```
   70 CONTINUE
```

begins the event description loop.

The statements

```
C
C3         READ AND WRITE EVENT DESCRIPTION
C
      READ(5,1010,END=840) IEVENT,T2,NC,(WEAPN(I),I=1,7),MDT
      WRITE(6,1270)
      DO 71 N=1,3
      CN(N)=OFF(N)
      IF(NC.NE.0) CN(N)=DEF(N)
   71 CONTINUE
      WRITE(6,1020) IEVENT, (CN(N),N=1,3), T2, (WEAPN(I),I=1,7), MDT
```

are used to read and write the event description (one of many); event number, time, type (defensive or offensive), weapons description, and mission descriptor defining the environmental conditions at the time of the event.

```
      IF(NC.NE.7) WRITE(6,1060)
        IF(NC.EQ.7) WRITE(6,1065)
```

are used to select between mode logic output table headings corresponding to electronics modes (NC = 0 to 6) or vehicle mode (NC = 7).

The statements

```
C
C4        READ SUBFUNCTIONAL FLOW FOR EVENT AND GENERATE SUBSYSTEM
C                  REQUIREMENTS VECTOR
C
          DO 250 I=1,L40
          DO 240 K=1,L1
          MD(I,K)=.FALSE.
          READ(5,1070)LF(I),LLF(K),LQ(I,K),LLQ(I,K),(FNAME(I,K,J),J=1,10),
         *             MQ(I,K),MMQ(I,K),MD(I,K),(LMAT(I,K,J),J=1,INCEI)
          L=0
          DO 90 J=1,INCEI
          IF(LMAT(I,K,J).EQ.0)GO TO 100
     80   L=L+1
          IF(LMAT(I,K,J).EQ.L)GO TO 90
          LMATT(L)=0
          GO TO 80
     90   LMATT(L)=1
          GO TO 140
    100   J=L+1
          IF(J.GT.INCEI)GO TO 140
          DO 110 L=J,INCEI
    110   LMATT(L)=0
    140   DO 150 L=1,INCEI
    150   LMA(I,K,L)    =LMATT(L)
```

are used to read the subsystem requirements, specified in terms of subsystem ordinal numbers, and then to derive the subsystem requirements binary vector for each subfunction and mode, and store it in the LMA array.

The statements

```
C
C5        WRITE SUBFUNCTIONAL FLOW
C
      IF(NC.NE.7)
     1WRITE(6,1080)LF(I),LLF(K),LQ(I,K),LLQ(I,K),(FNAME(I,K,J),J=1,10),
     *             MQ(I,K),MMQ(I,K),MD(I,K),(LMAT(I,K,J),J=1,INCEI)
      IF(NC.EQ.7) WRITE(6,1085) LF(I),LLF(K),LQ(I,K),LLQ(I,K),
     *  (FNAME(I,K,J),J=1,10),(LMAT(I,K,J),J=1,INCEI)
```

```
      IF(LLF(K)    .EQ.99) GO TO 230
      IF(LF(I)    .EQ.999) GO TO 260
      GO TO 240
230   LEE(I)=K-1
      GO TO 250
240   CONTINUE
250   CONTINUE
      I=L40+1
      READ(5,1070)M
      IF(M.EQ.999) GO TO 260
      WRITE(6,1090)
      STOP
```

are used to write the input subsystem requirements in binary form, along with subfunction, mode and mode logic, as shown in Figure 23.

The process of reading, generating the LMA vector, and writing as discussed in the previous two paragraphs continues until one of two things happen. If the event input list has been exhausted, control will be transferred to the statement "260 CONTINUE". However, if the number of inputs exceed the dimensional values, the program will write an error message and stop.

The statements

```
C
C6          INITIALIZE FOR MODE SEQUENCE IDENTIFICATION
C
 260  LE=I-1
      LQSVL=0
      LLQS=0
      NZ=0
      DO 290 J=1,L40
      DO 290 K=1,L1
      KS(J,K)=0
      DO 290 I=1,L72
      JES(J,K,I)=0
 290  CONTINUE
      DO 300 I=1,LCEI
      KW(I)=0
 300  CONTINUE
      DO 310 JCOUNT=1,256
      IJN(JCOUNT)=0
 310  CONTINUE
      JCOUNT=0
      JCAP=0
```

are used to initialize all variables used in the mode sequence identification process which starts in the next block of code.

The statements

```
C
C 7        GENERATE FINAL STATE OF THE SYSTEM - BINARY VECTOR KW
C
  330  CONTINUE
       DO 340 I=1,INCEI
       KW(I)=1
  340  CONTINUE
       IF(NZ.LT.1)GO TO 400
       LL1=1
  350  KW(LL1)=0
       IF(NZ.LT.2)GO TO 400
       LL2=LL1+1
  360  KW(LL2)=0
       IF(NZ.LT.3)GO TO 400
       LL3=LL2+1
  370  KW(LL3)=0
       IF(NZ.LT.4)GO TO 400
       LL4=LL3+1
  380  KW(LL4)=0
       IF(NZ.LT.5)GO TO 400
       LL5=LL4+1
  390  KW(LL5)=0
       IF(NZ.LT.6) GO TO 400
       LL6=LL5+1
   391 KW(LL6)=0
       IF(NZ.LT.7) GO TO 400
       LL7=LL6+1
   392 KW(LL7)=0
       IF(NZ.LT.8) GO TO 400
       LL8=LL7+1
   393 KW(LL8)=0
  400  CONTINUE
```

work in conjunction with a later block of code to generate a set of current state (binary) vectors KW, limited by the current number of zeros (NZ) in the array.

The statements

```
C
C 8        TEST FOR SUPPRESSED SUBSYSTEMS
C
      DO 410 MM=1,INCEI
      IF(MI(MM).NE.0)GO TO 410
      IF(KW(MM).EQ.1)GO TO 720
 410  CONTINUE
      IF(NC.EQ.7)GO TO 405
      IF(MCR.EQ.0)GO TO 402
      IF(KW(MCR).EQ.0)GO TO 720
  402 IF(MPR.EQ.0)GO TO 404
      IF(KW(MPR).EQ.0)GO TO 720
  404 IF(MAV.EQ.0)GO TO 405
      IF(KW(MAV).EQ.0)GO TO 720
  405 CONTINUE
      JCOUNT =JCOUNT+1
```

are used to reject (i.e., to withhold from further use) a state if it requires either:

1. A subsystem be viable that has been deleted (by the MI flag)

2. A subsystem (one of three) be failed that is critical to all significant modes (the subsystems are labeled MCR, MPR, and MAV).

The second test applies only to the electronic mode (NC = 0 to 6). Note that no number is assigned to a rejected state.

The statements

```
C
C 9        DEFINE MODE SEQUENCE AND SUBSYSTEMS USED
C
      DO 530 I=1,LCEI
 530  KU(I)=0
      LTEST=1
      DO 640 L = 1,LE
      LEQ=LEE(L)
      DO 540 K= 1,LEQ
 540  KS(L,K)=0
      IF(L.EQ.LQSVL.OR.LTEST.EQ.1) GO TO 550
      GO TO 640
 550  DO 630 K=1,LEQ
      IF(K.GE.LLQS.OR.LTEST.EQ.1) GO TO 560
      GO TO 630
 560  IF(.NOT.MDT.AND.MD(L,K)) GO TO 580
      GO TO 590
 580  LQSVL=MQ(L,K)
      LLQS=MMQ(L,K)
      LTEST=0
      IF(LQSVL.EQ.L) GO TO 630
      GO TO 640
 590  LTEST=1
      DO 600 M=1,INCEI
      IF(KW(M).LT.LMA(L,K,M))   GO TO 620
 600  CONTINUE
      DO 610 M=1,INCEI
      IF(KW(M).GT.LMA(L,K,M)   .OR.KW(M).EQ.0) GO TO 610
      IF(KW(M).EQ.LMA(L,K,M)   .AND.KW(M).EQ.1) KU(M)=1
 610  CONTINUE
      KS(L,K)=1
      LQSVL=LQ(L,K)
      LLQS=LLQ(L,K)
      LTEST=0
 620  IF(LQSVL.GT.L)GO TO 640
 630  CONTINUE
 640  CONTINUE
      J=JCAP+1
```

are used to generate the KS and KU binary arrays which define subfunction/mode numbers used and the subsystem numbers used, respectively, by the current state. LE and LEE are the number of subfunctions and modes (per subfunction) defined, which limit the outer (subfunction) and inner (mode) do-loops ending at 640 and 630, respectively. LQSVL and LLQS are the subfunction and mode numbers which are assigned for subsequent testing of the KW. LQSVL and LLQS are initially 1 and 1. They are later assigned the values MQ, MMQ if the mission descriptor (MD) test fails, and are assigned the values LQ, LLQ if the LMA test (subsystem requirements) is successful. The LMA test takes place in the inner do-loop on M, ending at 600. The KU array and KS array are loaded with a 1 after each successful mode. LTEST=0 is a flag denoting failure of the MD test, which results in rejection of further tests on the current subfunction and mode. When LMA test fails, the very next mode is required, so LTEST is set to 1, preventing the selection of the next subfunction. As a result of these processes, the KU (M) array is the union of all subsystem requirements of the mode sequence.

The statements

```
C
C10       ASSIGN MODE SEQUENCE NUMBERS
C
 650  J=J-1
      IF(J.NE.0) GO TO 680
      JCAP=JCAP+1
      J=JCAP
      DO 660 L = 1,LE
      LEQ=LEE(L)
      DO 660 K=1,LEQ
      JES(L,K,J)=KS(L,K)
 660  CONTINUE
      DO 670 IU=1,INCEI
      JU(IU,J)=KU(IU)
 670  CONTINUE
      GO TO 700
 680  DO 690 L = 1,LE
      LEQ=LEE(L)
      DO 690 K=1,LEQ
      IF(JES(L,K,J).NE.KS(L,K)) GO TO 650
 690  CONTINUE
 700  CONTINUE
      IJN(JCOUNT)=J
```

are used to assign a mode sequence number (J) to the current state number (JCOUNT). This is done by comparing the new JS array against its immediate predecessor, identified by the variable JES. If identical, the decremented J number is assigned to the current JCOUNT. If not, the next earlier KS is tested (i.e., J is decremented again). This is continued, if no match is found, to the first KS. If it still is not matched, the tentative JCAP is adopted as the J value. Thus, in the testing against all previous KS, any match stops the process at some J, which then gets ascribed to the current JCOUNT, along with the associated JES and JU (the final array representing subsystems used). But if it fails, the JCAP value, which has been

saved as a potential value of J from the beginning of the test, is finally authorized. Incidentally, JCAP tracks the largest value of J, and is used as a limit on do-loops in subsequent portions of the program.

The statements

```
C
C 11     REPOSITION ZEROS IN STATE BINARY VECTOR (KW)
C
  720 IF(NZ.LT.1.OR.NZ.GT.8)GO TO 790
      GO TO (770,760,750,740,730,728,726,724), NZ
  724 KW(LL8)=1
      LL8=LL8+1
      IF((LL8-8).LE.(INCEI-NZ)) GO TO 393
  726 KW(LL7)=1
      LL7=LL7+1
      IF((LL7-7).LE.(INCEI-NZ)) GO TO 392
  728 KW(LL6)=1
      LL6=LL6+1
      IF((LL6-6).LE.(INCEI-NZ)) GO TO 391
  730 KW(LL5)=1
      LL5=LL5+1
      IF((LL5-5).LE.(INCEI-NZ))GO TO 390
  740 KW(LL4)=1
      LL4=LL4+1
      IF((LL4-4).LE.(INCEI-NZ))GO TO 380
  750 KW(LL3)=1
      LL3=LL3+1
      IF((LL3-3).LE.(INCEI-NZ))GO TO 370
  760 KW(LL2)=1
      LL2=LL2+1
      IF((LL2-2).LE.(INCEI-NZ))GO TO 360
  77C KW(LL1)=1
      LL1=LL1+1
      IF((LL1-1).LE.(INCEI-NZ))GO TO 350
  790 CONTINUE
      NZ=NZ+1
      IF(NZ.LE.NZT)GO TO 330
      NZ=0
```

are used to increment the locations (LL1, LL2, etc.) of the (up to) eight possible zeros within the KW array after each state has been "pushed" through the preceding three blocks of code.

The statements

```
C
C 12     WRITE MODE SEQUENCE AND SUBSYSTEMS USED
C
      DO 820 J=1,JCAP
      WRITE(6,1170)
      WRITE(6,1140) J
      DO 800 L = 1,LE
```

```
      LEQ=LEE(L)
      DO 800 K=1,LEQ
      IF(JES(L,K,J).EQ.0) GO TO 800
      WRITE(6,1150) (FNAME(L,K,I   ),I=1,10)
  800 CONTINUE
      IF(NC.NE.7) WRITE(6,1180)
      IF(NC.EQ.7) WRITE(6,1185)
      DO 810 M=1,INCEI
      IF(JU(M,J).NE.1) GO TO 810
      WRITE(6,1190) (CNAME(K,M),K=1,LCN8)
  810 CONTINUE
  820 CONTINUE
```

are used to print the modes and subsystems associated with each mode sequence number. This is accomplished by means of the JES and JU arrays, associated with J.

The statements

```
C
C13         WRITE MODE SEQUENCE ON TAPE FOR USE IN PROGRAM 2
C
      WRITE(4,1280) (IJN(IJK),IJK=1,JCOUNT)
      GO TO 70
  840 END FILE 4
      REWIND 4
      STOP
```

write the mode sequence number array on unit 4. Control is transferred back to statement "70 CONTINUE" where another event is processed. When all events have been processed, the tape containing the mode sequence number is rewound and Program 1 stops.

The statements

```
C
C14         DEFINE FORMATS
C
  950 FORMAT(10I3)
  960 FORMAT (26H1       SYSTEM CONFIGURATION)
  970 FORMAT(1H0,8X,9HEQUIPMENT,25X, 4HMTBF,7X,7HTHDM(1),6X,7HTHDM(2),
     *7X,7HTHDM(3),8X,7HTHDM(4)/56X,3HG D,11X,1HN,12X,1HB,15X,1HT//1X)
  980 FORMAT(I3,1X,8A4,F10.2/4E12.5,T65,A2)
  990 FORMAT(1H ,1X,I3,2X,8A4,F10.2,2X,4(E12.5,2X),3X,A2)
 1000 FORMAT (50H0  TOO MANY INPUTS OR M IS OUTSIDE ALLOWABLE RANGE)
 1010 FORMAT(I3,1X,F6.2,1X,I3,1X,7A4,2X,L1)
 1020 FORMAT(1H0,5X,9HEVENT NO.,I2,1X,3HIS ,3A4/5X,15HEVENT OCCURRED ,
     *F6.2,
     *20H HOURS AFTER TAKEOFF/5X,21HEVENT DESCRIPTION IS ,7A4/5X,L1)
 1060 FORMAT (1H0,7X,16HSUBFUNCTION/MODE,32X,21HEQUIPMENT DESCRIPTION,12
     *          X,18HMISSION DESCRIPTOR/1X)
 1065 FORMAT(1H0, 7X,16HSUBFUNCTION/MODE,32X,35HFLIGHT AND DETECTION TIM
     *E REMAINING)
 1070 FORMAT(4I3,1X,10A4,1X,2I3,1X,L1/23I3)
 1080 FORMAT(1X,2I3,2X,2I3,2X,10A4,T86,2I3,12X,L1,T59,23I1)
```

```
1085 FORMAT(1X,2I3,2X,2I3,2X,10A4,                       T59,23I1)
1C90 FORMAT (29HO TOO MANY INPUTS FOR F ARRAY)
1140 FORMAT(1HO,11X,I3)
115C FORMAT(1H ,45X,10A4)
1170 FORMAT(1H ,5X,16HMODE SEQUENCE NO,16X,18HSUBFUNCTIONAL FLOW)
1180 FORMAT(1HO,58X,15HSUBSYSTEMS USED/1X)
1185 FORMAT(1HO,58X,31HACTUAL AND APPARENT FLIGHT TIME)
1190 FORMAT(1H ,60X,8A4)
1270 FORMAT(23H1        EVENT DESCRIPTION)
1280 FORMAT(25I3)
     END
```

define the formats used by Program 1 to read and write the inputs and outputs.

PROGRAM 2

Program 2 consists of a main routine and one subroutine. The purpose of Program 2 is to use the mode sequences defined in Program 1 in conjunction with input capabilities (own system and threat) to compute the mode probabilities and system effectiveness and survivability parameters.

Main Routine

The purpose of the main routine is to control the inputs and perform all logic and computations except for an interpolation that is performed by the subroutine.

The statements

```
C ***********************************************************************
C
C        MISSION DAMAGE EFFECTIVENESS MODEL - PROGRAM TWO
C                               MISDEM - PGM 2
C
C ***********************************************************************
      DIMENSION CEP(50),CN(3),CURVE(4,30),DEF(3),DNAME(8),FA(50),
     1  FTDNN(16,16),FTFNN(16),GRAPH(30),HDMT(4),ISUB(4),IT(23),ITT(23)
     2  ,IV(23),IZ(23),JZ(23),KW(23),KWW(23),MI(23) ,MTBF(23),OFF(3),
     3  PCAP(50),PCKILL(23,10,27),PCSURV(23,10,27),PI(256),PJ(256),
     4  PK(50),PM(23),PMM(23),PMISS(27,256),PTDN(16,16),PTFN(16),
     5  QO(256),QPRM(23,256),R(10),RA(23),RESUL(2),RM(24),THDM(23,4),
     6  TIMEF(23),TIMEN(23),WEAPN(7)
      INTEGER*2 IJN(256)
      LOGICAL MDT
      DATA OFF/4HOFFE,4HNSIV,4HE   /,DEF/4HDEFE,4HNSIV,4HE   /,OUT/2HNO/
```

are used to allocate storage by use of the dimension statement to reserve two integer storage locations for each of the 256 INTEGER*2 words, declare the variable MDT as logical with four storage locations allocated for each, and define the initial values for various variables by use of the data statements.

The statements

```
C
C1        INITIALIZE
C
      LCN8=8
      LCEI=23
      L72=40
```

are used to define variable names for the dimensional constants. LCN8 is the maximum number of equipment name segments. LCEI is the maximum number of subsystems in the equipment configuration, used as a check (later in the program) on program input. L72 is the maximum number of mode sequences generated by Program 1 for any event.

The statements

```
      INCEI=0
      DO 30 I=1,LCEI
      MTBF(I)=0.0
      PM(I)=0.0
      MI(I)=1
      IV(I)=I
      IZ(I)=I
      DO 20 MPRM=1,4
      THDM(I,MPRM)=0
   20 CONTINUE
   30 CONTINUE
```

are used to initialize the subsystems counter (INCEI) to zero, to set all MTBF values and probability of reliable operation values equal to zero, to set the on-off flag for all subsystems to "on", to initialize the event and subsystem counter arrays and to zero-out all damage threshold storage locations for all subsystems.

The statement

```
       READ (5,1110) NZT,MCR,MPR,MAV,MLTH,NABORT
```

is used to read the first input data card. These variables are used for control of state generation (through NZT), state probability generation (through MCR, MPR, and MAV), mission length (MLTH), and event abort control (NABORT).

The statements

```
C
C2        READ AND WRITE SYSTEM DESCRIPTION
C
      WRITE (6,1120)
      WRITE (6,1130)
      DO 50 I=1,LCEI
      READ (5,1140) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4),TMN
     1,TMF,ONOFF
      IF (M.EQ.999) GO TO 70
      WRITE (6,1150) M,(DNAME(J),J=1,LCN8),TBFM,(HDMT(MPRM),MPRM=1,4)
      IF (M.GT.LCEI.OR.M.LT.1) GO TO 60
      IF (M.GT.INCEI) INCEI=I
      IF (ONOFF.EQ.OUT) MI(I)=0
      DO 40 MPRM=1,4
      THDM(I,MPRM)=HDMT(MPRM)
   40 CONTINUE
      MTBF(I)=TBFM
      TIMEN(I)=TMN
      TIMEF(I)=TMF
   50 CONTINUE
      READ (5,1140) M
      IF (M.EQ.999) GO TO 70
   60 WRITE (6,1160)
      STOP
   70 CONTINUE
```

are used for input/output of the system description. The first two lines write the headings in preparation for the output.

The next statement starts a loop which allows for the maximum input of 23 subsystem definitions. Each subsystem definition consists of the equipment names, MTBF, nuclear damage thresholds and a logical flag (ON OFF) which can be used to suppress a subsystem (prevent its being considered further). If the system number that is input is equal to 999, the program switches control out of the loop (i.e., "70 CONTINUE") signifying there are no more inputs of this type. However, if a system configuration is input, the program proceeds to write out information. The next statement is used to check for blanks (subsystems left out) or too many subsystems (relative to LCEI) in the input data. If either of these conditions exist, program control is transferred to an error message which is written out and the program stops. The next statement updates the counter which keeps track of the total number of subsystems input.

The next statement is used to set a flag if a subsystem is being suppressed. The internal do-loop which follows stores the damage threshold information with the subsystem index. The next three statements take the MTBF, the time on and time off information and stores them by the subsystem index.

Program control is then transferred to the beginning of the system configuration loop to read the next subsystem description. Should the program go through the loop 23 times and never transfer control to the statement which says "70 CONTINUE", the program will read another card. If this card contains another system description, the program will write an error message and stop. However, if the input contains a "999" for the subsystem number, control will be transferred to the "70 CONTINUE" statement and program execution will flow as described in the next set of statements.

The statements

```
C
C3        INITIALIZE
C
      ET=0.0
      II=0
      T1=0.0
```

initialize the values of expected number of targets killed, the previous event type, and event time.

The statements

```
      REWIND 4
      REWIND 3
```

rewind the tapes on units three and four so that they will start at the beginning.

The statements

```
C
C4        INITIALIZE
C
   80 CONTINUE
      DO 85 M=1,INCEI
      RM(M)=0.0
   85 CONTINUE
```

begin the event description loop and zero out the lethal radius storage locations for all subsystems.

The statements

```
C
C5        READ AND WRITE EVENT DESCRIPTION
C
      READ (5,1170,END=1100) IEVENT,T2,NC,(WEAPN(I),I=1,7),MDT
      WRITE (6,1230)
      DO 90 N=1,3
      CN(N)=OFF(N)
```

```
      IF (NC.NE.0) CN(N)=DEF(N)
   90 CONTINUE
      WRITE (6,1180) IEVENT,(CN(N),N=1,3),T2,(WEAPN(I),I=1,7),MDT
```

are used to read and write the event description (one of many): event number, time, type (defensive or offensive), weapon description, and a mission descriptor defining the environmental conditions at the time of the event.

The statements

```
      DO 100 L=1,INCEI
      IZ(L)=L
  100 CONTINUE
```

are used to copy the subsystems ordinal number into the vulnerability index array as initial values which are changed later, only if there is a nuclear event.

The statement

```
      IF (NC.EQ.0.OR.NC.EQ.6 .OR. NC.EQ.7) GO TO 200
```

transfers control to a later section of the program if the event is not an electronics mode nuclear-definsive event.

The next set of statements

```
C
C6       COMPUTE LETHAL RADII AND ASSIGN INDEX
C
      DO 120 I=1,NC
      READ (5,1240) NPOINT
      N2=2*NPOINT
      READ (5,1220) ISUB(I),(CURVE(I,J),J=1,N2)
      IS=ISUB(I)
      DO 120 M=1,INCEI
      DO 110 IJ=1,N2
      GRAPH(IJ)=CURVE(I,IJ)
  110 CONTINUE
      CALL TWODNZ (THDM(M,IS),1,GRAPH,NPOINT,2,RESUL)
      RA(M)=RESUL(2)
      RM(M)=AMAX1(RA(M),RM(M))
  120 CONTINUE
      INE=INCEI+1
      RM(INE)=0.0
      DO 140 LOO=1,INCEI
      INIT=LOO+1
      DO 130 J=INIT,INE
      IF (RM(J).LE.RM(LOO)) GO TO 130
      STORE=RM(J)
```

```
      RM(J)=RM(LOO)
      RM(LOO)=STORE
      ITORE=IZ(LOO)
      IZ(LOO)=IZ(J)
      IZ(J)=ITORE
  130 CONTINUE
  140 CONTINUE
      WRITE (6,1250) (RM(M),M=1,INE)
```

will be discussed in two parts in order to clarify their use. These statements are used for nuclear defensive events only.

The first set encompasses statements from "DO 120" through "120 CONTINUE". The statements through "110 CONTINUE" are used to read nuclear weapon effect data and store the information in GRAPH for four damage mechanisms. The CALL TWODNZ applies a linear interpolation routine to derive the lethal radius associated with each subsystem and each of the four damage mechanisms.

The two statements following the CALL statement detect the maximum value of the lethal radius of the subsystem, over all four damage mechanisms.

The second set of statements beginning with INE = INCEI + 1 are used to reassign the lethal radii in descending order and to assign a vulnerability index number (IZ) to each subsystem, where 1 corresponds to the maximum lethal radius. The lethal radii values are subsequently printed out.

The next statements

```
C
C7        COMPUTE SUBSYSTEM RELIABILITY IN TRANSITION
C
  200 CONTINUE
      DO 230 M=1,INCEI
      PM(M)=1.0
      IF (MI(M).EQ.0) GO TO 230
      IF (T2.LE.TIMEN(M)) GO TO 230
      IF (T2.GE.TIMEF(M)) GO TO 210
      DELTAT=T2-T1
      IF (T1.LE.TIMEN(M)) DELTAT=T2-TIMEN(M)
      GO TO 220
  210 DELTAT=0.0
      IF (TIMEF(M).GT.T1) DELTAT=TIMEF(M)-T1
  220 PM(M)=1.0-DELTAT/MTBF(M)+DELTAT**2/(2.0*MTBF(M)**2)
  230 CONTINUE
      T1=T2
      DO 240 L=1,INCEI
      JV=IV(L)
      PMM(L)=PM(JV)
  240 CONTINUE
```

are used to compute the reliability of every subsystem which is time dependent. The reliability equation is:

$$d_k = \exp(-\Delta t \div MTBF) = 1 - \Delta t/MTBF + \Delta t^2/2MTBF^2$$

where

d_k represents subsystem reliability in the transition

Δt is the elapsed subsystem time in the transition

MTBF is the mean time between failures

The statements

```
C
C8      INITIALIZE STATE AND STATE PROBABILITIES
C
        NZ=0
        NZI=0
        DO 250 I=1,LCEI
        KW(I)=0
  250   CONTINUE
        DO 260 J=1,L72
        PCAP(J)=0.0
  260   CONTINUE
        JCAP=0
        IF (IEVENT.NE.1) GO TO 290
        ICOUNT=1
        DO 270 M=1,INCEI
        IT(M)=1
        ITT(M)=1
        IF (MI(M).EQ.1) GO TO 270
        IT(M)=0
        ITT(M)=0
  270   CONTINUE
```

initialize the current state binary vector and probability of all mode sequences at zero. If the event is not the first, program control is transferred to a later part of the program. However, if it is the first event, the prior state vectors IT and ITT are initialized to all "ones" (except deleted subsystems) representing a perfect state at takeoff and no other prior states are considered for this event.

The statements

```
C
C9         ABORT FLOW CONTROL
C
        IF(NABORT.NE.0)GO TO 275
        GO TO 278
```

differentiate between abort events (used for simulating the first event of an abort mission) and non-abort events and directs the program control accordingly.

For those first abort mission events, the statements

```
C
C10        INITIALIZE ABORT STATES
C
  275 IF (I EVENT.NE.1) GO TO 290
      READ(3)JEVENT,JCOUNT,(PJ(L),L=1,JCOUNT)
      IF(JEVENT.NE.NABORT)GO TO 275
      DO 277 I=1,JCOUNT
      PI(I)=PJ(I)
  277 PJ(I)=0.0
      GO TO 305
```

read the state probability on tape unit 3 into prior state probability PI for the event number indicated by (equal to) NABORT. In addition, the current state probability PJ is initialized to zero. Program control is transferred to a later section of the program which generates the initial state of the system.

The statements

```
C
C11        INITIALIZE NORMAL STATES
C
  278 DO 280 L=1,256
      PI(L)=1.0
      PJ(L)=0.0
  280 CONTINUE
      GO TO 310
  290 DO 300 I=1,JCOUNT
      PI(I)=PJ(I)
      PJ(I)=0.0
  300 CONTINUE
  305 CONTINUE
      ICOUNT=0
```

initialize the non-abort state probabilities for the prior event (PI) equal to one and for the current event (PJ) equal to zero. Since the event is not an abort event, the prior event probabilities are set equal to the current event probabilities from the last event, and the current event probabilities are reinitialized to zero. The prior state counter is also set equal to zero.

131

The statements

```
C
C12       GENERATE A PRIOR STATE OF THE SYSTEM (IT)
C
  310 IF(NABORT.EQ.0)GO TO 315
      GO TO 320
315   CONTINUE
      IF(IEVENT.EQ.1)GO TO 440
  320 CONTINUE
      DO 330 I=1,INCEI
      IT(I)=1
  330 CONTINUE
      IF (NZ.LT.1) GO TO 410
      II1=1
  360 IT(II1)=0
      IF (NZ.LT.2) GO TO 410
      II2=II1+1
  370 IT(II2)=0
      IF (NZ.LT.3) GO TO 410
      II3=II2+1
  380 IT(II3)=0
      IF (NZ.LT.4) GO TO 410
      II4=II3+1
  390 IT(II4)=0
      IF (NZ.LT.5) GO TO 410
      II5=II4+1
  400 IT(II5)=0
      IF(NZ.LT.6)GO TO 410
      II6=II5+1
  405 IT(II6)=0
      IF(NZ.LT.7)GO TO 410
      II7=II6+1
  406 IT(II7)=0
      IF(NZ.LT.8)GO TO 410
      II8=II7+1
  407 IT(II8)=0
```

are used to place exactly NZ zeros into the prior state vector. This block of code works with the block of code just prior to Compute State Probabilities, to generate all prior states.

The statements

```
C
C13       TEST FOR DELETED OR CRITICAL SUBSYSTEMS
C
  410 CONTINUE
      DO 420 MM=1,INCEI
      IF (MI(MM).EQ.1) GO TO 420
      IF (IT(MM).EQ.1) GO TO 840
  420 CONTINUE
      IF(NC.EQ.7) GO TO 425
      IF(MCR.EQ.0)GO TO 422
```

```
      IF(IT(MCR).EQ.0) GO TO 840
  422 IF(MPR.EQ.0)GO TO 424
      IF(IT(MPR).EQ.0) GO TO 840
  424 IF(MAV.EQ.0)GO TO 425
      IF(IT(MAV).EQ.0) GO TO 840
  425 ICOUNT=ICOUNT+1
```

are used to test for deleted and critical subsystems. This block of code is identical to the one in Program 1.

The statements

```
C
C14        RESHUFFLE SUBSYSTEM ORDER
C
      DO 430 L=1,INCEI
      JV=IV(L)
      ITT(L)=IT(JV)
  430 CONTINUE
  440 JCOUNT=0
```

reorder the subsystems in preparation for use in the transition algorithm.

The statements

```
C
C15        GENERATE A CURRENT STATE OF THE SYSTEM (KW)
C
  450 CONTINUE
      DO 460 I=1,INCEI
      KW(I)=1
  460 CONTINUE
      IF (NZI.LT.1) GO TO 540
      LL1=1
  490 KW(LL1)=0
      IF (NZI.LT.2) GO TO 540
      LL2=LL1+1
  500 KW(LL2)=0
      IF (NZI.LT.3) GO TO 540
      LL3=LL2+1
  510 KW(LL3)=0
      IF (NZI.LT.4) GO TO 540
      LL4=LL3+1
  520 KW(LL4)=0
      IF (NZI.LT.5) GO TO 540
      LL5=LL4+1
  530 KW(LL5)=0
      IF(NZI.LT.6)GO TO 540
      LL6=LL5+1
```

```
  535 KW(LL6)=0
      IF(NZI.LT.7)GO TO 540
      LL7=LL6+1
  536 KW(LL7)=0
      IF(NZI.LT.8)GO TO 540
      LL8=LL7+1
  537 KW(LL8)=0
  540 CONTINUE
```

are completely analogous to the initial state generation and are identically coded.

The statements

```
C
C16        TEST FOR SUPPRESSED SUBSYSTEMS
C
      DO 550 MM=1,INCEI
      IF (MI(MM).EQ.1) GO TO 550
      IF (KW(MM).EQ.1) GO TO 710
  550 CONTINUE
      IF(NC.EQ.7)GO TO 555
      IF(MCR.EQ.0)GO TO 552
      IF(KW(MCR).EQ.0)GO TO 710
  552 IF(MPR.EQ.0)GO TO 554
      IF(KW(MPR).EQ.0)GO TO 710
  554 IF(MAV.EQ.0)GO TO 555
      IF(KW(MAV).EQ.0)GO TO 710
  555 JCOUNT=JCOUNT+1
```

are completely analogous to the prior state test and are identically coded.

The statements

```
C
C17        RESHUFFLE SUBSYSTEM ORDER
C
      DO 560 L=1,INCEI
      JV=IV(L)
      KWW(L)=KW(JV)
  560 CONTINUE
```

are completely analogous to the prior state subsystem reordering and are identically coded.

The statement

```
C
C18       COMPUTE THE SYSTEM STATE TRANSITION PROBABILITY
C
      IF (II.NE.6) GO TO 1500
```

allows the first transition algorithm to be used, which applies to electronics mode quick conventional threat type events. The test parameter is II, the prior event type.

The statements

```
C
C 19     QUICK CONVENTIONAL THREAT DAMAGE AND RELIABILITY
C
      SUMTRK=0.0
      DO 620 K=1,KMAX
      SUMTRA=0.0
      DO 610 L=1,LMAX
      TRANS=1.0
      DO 600 M=1,INCEI
      PCSURV(M,L,K) = 1.0 - PCKILL(M,L,K)
      IF (ITT(M)-KWW(M)) 570,580,590
  570 TRANS=0.0
      GO TO 620
  580 IF (ITT(M).EQ.0.0) GO TO 600
      TRANS=PCSURV(M,L,K)*PMM(M)*TRANS
      GO TO 600
  590 TRANS=(1.0-PCSURV(M,L,K)*PMM(M))*TRANS
  600 CONTINUE
      SUMTRA=TRANS+SUMTRA
  610 CONTINUE
      SUMTRK=(SUMTRA/LMAX)*PMISS(K,ICOUNT)+SUMTRK
  620 CONTINUE
      TRANS=SUMTRK
      GO TO 700
 1500 IF(II.NE.7) GO TO 630
```

compute the transition probability TRANS for the state-pair under consideration. The last statement transfers control to the nuclear transition algorithm if appropriate. Otherwise, the prior event is deemed to be a vehicle mode application, which requires the slow threat damage transition algorithm which follows.

The statements

```
C
C 20     SLOW THREAT DAMAGE
C
      TRANS = 1.0
      NTFN=0
      ITFN=0
      IINCEI=INCEI/2
      DO 1510 L=1,IINCEI
      IF(KWW(L).EQ.1)NTFN=2**((INCEI/2)-L)+NTFN
 1510 IF(ITT(L).EQ.1)ITFN=2**((INCEI/2)-L)+ITFN
      NTDN=0
      ITDN=0
      JINCEI=IINCEI+1
      DO 1520 L=JINCEI,INCEI
      IF(KWW(L).EQ.1)NTDN=2**(INCEI-L)+NTDN
```

```
1520 IF(ITT(L).EQ.1)ITDN=2**(INCEI-L)+ITDN
     NTTF=NTFN+1
     ITTF=ITFN+1
     NTTD=NTDN+1
     ITTD=ITDN+1
     IF(ITTD.GT.ITTF.OR.NTTD.GT.NTTF)GO TO 1605
     IF(ITTF.LE.1.OR.ITTD.LE.1) GO TO 1605
     IF(NTTF.GT.(ITTF-1).OR.NTTD.GT.(ITTD-1))GO TO 1605
     GO TO 1610
1605 TRANS1=0
     GO TO 1720
1610 TRANS1=0
     IF(NTTF-(ITTF-1))1670,1620,1620
1620 IF(NTTD-(ITTD-1))1630,1625,1625
1625 DO 1627 I=NTTF,MLTH
     SUMPTD=0
     DO 1626 J=NTTD,I
1626 SUMPTD=PTDN(I,J)+SUMPTD
1627 TRANS1=PTFN(I)*SUMPTD+TRANS1
     GO TO 1720
1630 DO 1631 I=NTTF,MLTH
1631 TRANS1=PTFN(I)*PTDN(I,NTTD)+TRANS1
     GO TO 1720
1670 IF(NTTD-(ITTD-1))1680,1675,1675
1675 DO 1676 J=NTTD,NTTF
1676 TRANS1=PTDN(NTTF,J)+TRANS1
     TRANS1=PTFN(NTTF)*TRANS1
     GO TO 1720
1680 TRANS1=PTFN(NTTF)*PTDN(NTTF,NTTD)
1720 TRANS=TRANS1*TRANS
     GO TO 700
```

compute the transition probability for the vehicle mode. The statements through 1520 convert the binary t and τ values to decimal values. The next four statements change the time reference from the prior event to the current event. The next six statements eliminate unwanted states by setting the transition probability to zero. The remaining statements are as described in the flow chart (Figure 8). The four transition probability equations are given in Table 6.

The statements

```
C
C 21     QUICK NUCLEAR DAMAGE AND RELIABILITY
C          (ALSO NON-DEFENSIVE TRANSITIONS)
C
```

```
630 TRANS=1.0
    NF=1
    IF(II.EQ.0) GO TO 640
    TRANS=QO(ICOUNT)
    NF=0
640 CONTINUE
    DO 690 M=1,INCEI
    IF (ITT(M)-KWW(M)) 650,660,670
650 TRANS=0.0
    GO TO 700
660 IF (ITT(M).EQ.0) GO TO 680
    NF=1
    TRANS=TRANS*PMM(M)
    GO TO 690
670 TRANS=TRANS*(1.0-PMM(M))
680 IF (NF.EQ.0) TRANS=TRANS+QPRM(M,ICOUNT)
690 CONTINUE
```

compute the transition probability for the electronics nuclear events and non-defensive electronics events. The flag NF = 1 in the third statement indicates that nuclear damage is not considered. This is revised immediately if the prior event was electronics-nuclear, but is reset to 1 when the first surviving subsystem is encountered. For non-nuclear events, only reliability effects occur.

The statements

```
C
C22       COMPUTE THE STATE PROBABILITIES
C
  700 PJ(JCOUNT)=TRANS*PI(ICOUNT)+PJ(JCOUNT)
```

accumulate (add) the probabilities for all transitions into the same current state. This sum is the current state probability. This is repeated for all current states as the state do-loop is executed. The equation is:

$$P_j, n = \sum^{K} (P_{j/i,n})x(Pi,n)$$

The statements

```
C
C23       RESET CURRENT STATE (KW) ZERO LOCATIONS
C
  710 IF (NZI.LT.1.OR.NZI.GT.8) GO TO 820
      GO TO (800,780,760,740,720,715,714,713),NZI
  713 KW(LL8)=1
      LL8=LL8+1
      IF((LL8-8).LE.(INCEI-NZI))GO TO 537
```

```
714 KW(LL7)=1
    LL7=LL7+1
    IF((LL7-7).LE.(INCEI-NZI))GO TO 536
715 KW(LL6)=1
    LL6=LL6+1
    IF((LL6-6).LE.(INCEI-NZI))GO TO 535
720 KW(LL5)=1
    LL5=LL5+1
    IF((LL5-5).LE.(INCEI-NZI))GO TO 530
740 KW(LL4)=1
    LL4=LL4+1
    IF((LL4-4).LE.(INCEI-NZI))GO TO 520
760 KW(LL3)=1
    LL3=LL3+1
    IF((LL3-3).LE.(INCEI-NZI))GO TO 510
780 KW(LL2)=1
    LL2=LL2+1
    IF((LL2-2).LE.(INCEI-NZI))GO TO 500
800 KW(LL1)=1
    LL1=LL1+1
    IF((LL1-1).LE.(INCEI-NZI))GO TO 490
820 CONTINUE
    NZI=NZI+1
    IF (NZI.LE.NZT) GO TO 450
    NZI=0
```

work in conjunction with an earlier block of code called "Generate Prior State" to do just that. The role of these statements is to increment the location of the right-most zero in the state vector until it has gone as far as it can to the right and then increment the next zero location, etc. If all zeros have been moved as far as possible to the right, another zero is brought into play. However, if the added zero exceeds the number provided by the user (NZT test), the process of generating states is arrested and the program continues.

The statements

```
C
C24      ABORT FLOW CONTROL
C
    IF(NABORT.EQ.0)GO TO 835
    GO TO 840
835 CONTINUE
    IF (IEVENT.EQ.1) GO TO 970
```

control state generation for first events. If the event is not an abort and is the first event, there is only one initial state, and the prior state generation is bypassed. If the event is an abort, the normal state generation process is continued.

The statements

```
C
C25      RESET PRIOR STATE (IT) ZERO LOCATIONS
C
  840 IF (NZ.LT.1.OR.NZ.GT.8) GO TO 950
      GO TO (930,910,890,870,850,845,844,843), NZ
  843 IT(II8)=1
      II8=II8+1
      IF((II8-8).LE.(INCEI-NZ))GO TO 407
  844 IT(II7)=1
      II7=II7+1
      IF((II7-7).LE.(INCEI-NZ))GO TO 406
  845 IT(II6)=1
      II6=II6+1
      IF((II6-6).LE.(INCEI-NZ))GO TO 405
  850 IT(II5)=1
      II5=II5+1
      IF((II5-5).LE.(INCEI-NZ))GO TO 400
  870 IT(II4)=1
      II4=II4+1
      IF((II4-4).LE.(INCEI-NZ)) GO TO 390
  890 IT(II3)=1
      II3=II3+1
      IF((II3-3).LE.(INCEI-NZ)) GO TO 380
  910 IT(II2)=1
      II2=II2+1
      IF((II2-2).LE.(INCEI-NZ)) GO TO 370
  930 IT(II1)=1
      II1=II1+1
      IF((II1-1).LE.(INCEI-NZ))GO TO 360
  950 CONTINUE
      NZ=NZ+1
      IF (NZ.LE.NZT) GO TO 320
      NZ=0
  970 CONTINUE
```

are analogous to the "Reset Current State Zero Locations" block of code and is identically coded.

The statements

```
C
C26    WRITE OUTPUT STATE PROBABILITY TAPE
C
      IF(NABORT.NE.0)GO TO 979
      JEVENT=IEVENT
      WRITE(3) JEVENT,JCOUNT,(PJ(L),L=1,JCOUNT)
```

prepare a record of state probabilities on a normal mission, which may be employed to initialize an abort mission. The variable JEVENT is used to denote a normal mission event number so that the IEVENT variable may be used on an abort mission without ambiguity.

The statements

```
C
C27     COMPUTE THE MODE SEQUENCE PROBABILITIES
C
   979 READ (4,1280) (IJN(IJK),IJK=1,JCOUNT)
       JCAP=0
       DO 980 IJK=1,JCOUNT
       J=IJN(IJK)
       PCAP(J)=PCAP(J)+PJ(IJK)
       JCAP=MAX0(JCAP,J)
   980 CONTINUE
       PARIVE=PCAP(1)+PCAP(2)+PCAP(3)
```

are used to read in the mode sequence number array from program one and then distribute all state probability into variables PCAP(J) (representing probability of the Jth mode sequence). The distribution is made in accordance with the mode sequence number (J) array, whose subscript is the state number. PARIVE is the sum of probabilities of the first three mode sequences. In a vehicle mission, this sum is printed out (later). The mode sequence probability equation is:

$$P_{j,n} = \sum_{j=1}^{jmax} P_{j,n}$$

The statements

```
C
C28      READ AND WRITE OUTPUT MODE PROBABILITIES AND CAPABILITES
C
       DO 1000 J=1,JCAP
       READ (5,1260) CEP(J),FA(J),PK(J)
       IF (NC.EQ.0) GO TO 990
       IF(NC.NE.7) WRITE (6,1270) J,PCAP(J),CEP(J),FA(J)
       IF(NC.EQ.7) WRITE(6,1275) J,PCAP(J)
       GO TO 1000
   990 WRITE (6,1190) J,PCAP(J),PK(J)
  1000 CONTINUE
       IF(NC.NE.7) GO TO 9030
       WRITE(6,9020)PARIVE
  9020 FORMAT(1X,'PARIVE = ',E12.5)
```

are used to write the mode sequence probabilities, and read and write offensive and defensive capabilities as appropriate to the event type and MISDEM mode. If the current event is not a vehicle simulation, an additional set of statements are executed. For offensive events, the output effectiveness equations are:

$$P_{K,m} = \sum_{J=1}^{JCAP} P_{k,J} \times P_{J,n}$$

$$E_T(N) = \sum_{n=1}^{N} P_{K,n}$$

The statements

```
C
C 29      IF EVENT IS OFFENSIVE,COMPUTE ET - MISSION EFFECTIVENESS
C
 9030 II=0
      IF (NC.NE.0) GO TO 1030
      DO 1010 J=1,INCEI
      IV(J)=J
 1010 CONTINUE
      SUM=0.0
      DO 1020 J=1,JCAP
      SUM=SUM+PCAP(J)*PK(J)
 1020 CONTINUE
      WRITE (6,1200) IEVENT,SUM
      ET=ET+SUM
      WRITE (6,1210) ET
      GO TO 80
```

are used to set the prior event type variable II to zero in preparation of the next event, and remains so unless the current event is not. If offensive, the vulnerability indices are initialized equal to subsystem ordinal number, and cumulative expected number of targets killed is updated and printed out. If the event is not offensive, control is transferred to the next block of code.

The statements

```
C
C
C30      IF EVENT IS DEFENSIVE DETERMINE SYSTEM SURVIVABILITY PARAMETERS
C
C
 1030 IF(NC.EQ.7) GO TO 2500
      II=1
      DO 1040 IQ=1,INCEI
      IV(IQ)=IZ(IQ)
 1040 CONTINUE
      IF (NC.NE.6) GO TO 1070
      II=6
```

are used to set prior event type equal to 1 and vulnerability indices equal to the rank order dictated by this current nuclear/conventional defensive-electronics event.

The statements

```
C
C 31      QUICK CONVENTIONAL KILLS
C
      READ (5,1290) LMAX,KMAX,(R(K),K=1,KMAX)
      WRITE (6,1300) (R(K),K=1,KMAX)
      DO 1050 M=1,INCEI
      DO 1050 L=1,LMAX
      READ (5,1310) (PCKILL(M,L,K),K=1,KMAX)
```

```
1050 WRITE (6,1320) M,L,(PCKILL(M,L,K),K=1,KMAX)
     KMAX1=KMAX-1
     DO 1060 IJK=1,JCOUNT
     PMISS(KMAX,IJK)=1.0
     J=IJN(IJK)
     SIGMA=CEP(J)/1.178
     DENUM=2.0*SIGMA**2
     DEN175=175.0*DENUM
     EX1=1.0
     DO 1060 K=1,KMAX1
     EX2=0.0
     RM2=R(K)**2
     IF (RM2.LT.DEN175) EX2=EXP(-RM2/DENUM)
     PMISS(K,IJK)=EX1-EX2
     PMISS(KMAX,IJK)=PMISS(KMAX,IJK)-PMISS(K,IJK)
     EX1=EX2
1060 CONTINUE
     GO TO 80
```

are used first to load all subsystem kill probabilities associated with KMAX miss distances and LMAX offset angles, for use in computing the conventional weapon transition probability in the next event. The statements after 1050 are used to compute the probability of a threat warhead miss in a circular zone whose inner radius is R(K). The three equations used in this block of code are given below. The standard deviation (sigma) is derived from the threat weapon CEP as follows:

$$\sigma = CEP/1.178$$

The result of the integration of the bivariate normal distribution over limits R_k and $R_{k+1} > R_k$) is:

$$P_{miss}(K) = \exp\left(R_k{}^2/2\sigma^2\right) - \exp\left(R_{k+1}^2/2\sigma^2\right)$$

The transition probability associated with such an event is then computed as:

$$P_{j/i} = \sum_{k=1}^{K} P_{j/i}\text{ (given burst point k) x P (burst point k)}$$

The statements

```
C
C 32      SLOW KILLS
C
 2500 II=7
      DO 2122 I=1,MLTH
      READ(5,2120) FTFNN(I),(FTDNN(I,J),J=1,MLTH)
 2120 FORMAT(17F3.2)
      IFF=I-1
      IF(I.EQ.1)   GO TO 2125
```

```
      PTFN(I) = (FTFNN(I)-FTFNN(IFF))
      GO TO 2129
 2125 PTFN(I)=FTFNN(I)
 2129 CONTINUE
        DO 2130 J=1,MLTH
      JFF=J-1
      IF(J.EQ.1)GO TO 2140
      PTDN(I,J)=FTDNN(I,J)-FTDNN(I,JFF)
      GO TO 2130
 2140 PTDN(I,J) = FTDNN(I,J)
 2130 CONTINUE
 2122 CONTINUE
      GO TO 80
```

are used first to read in the flight time and abort detection time distribution functions. The statements after 2120 are used to compute the discrete probabilities of flight time and abort detection time for all values up to and including the duration of the mission MLTH. The flight time algorithm ends at 2129; the abort detection time algorithm ends at 2130.

The statements

```
C
C33        QUICK NUCLEAR KILLS
C
 1070 CONTINUE
      DO 1090 IJK=1,JCOUNT
      QO(IJK)=0.0
      J=IJN(IJK)
      SIGMA=CEP(J)/1.178
      DENUM=2.0*SIGMA**2
      DEN175=175.0*DENUM
      EX1=0.0
      RM2=RM(1)**2
      IF (RM2.LT.DEN175) EX1=EXP(-RM2/DENUM)
      QO(IJK)=EX1**FA(J)
      DO 1080 M=1,INCEI
      QPRM(M,IJK)=0.0
      RMM2=RM(M)**2
      RMM12=RM(M+1)**2
      IF (RMM2.LE.RMM12.OR.RMM12.GE.DEN175) GO TO 1080
      EXM1=EXP(-RMM12/DENUM)
      EXM=0.0
      IF (RMM2.LT.DEN175) EXM=EXP(-RMM2/DENUM)
      QPRM(M,IJK)=EXM1**FA(J)-EXM**FA(J)
 1080 CONTINUE
 1090 CONTINUE
      GO TO 80
```

are used to compute the probability of a nuclear weapon miss in the circular zone whose outer radius is RM (M). Since the miss is assumed normally distributed, the miss probability is negligibly small beyond a few standard deviations. This fact is used in setting up a test for RM to avoid underflow in the subtraction of two vanishing numbers in the fourth statement from the bottom. The equation for miss probability is:

$$q'_{k,i} = \exp(-SPdr_{k+1}^2/2\,\sigma_i^2) - \exp(-SPdr_k^2/2\,\sigma_i^2)$$

The following statements are used to rewind the tapes and stop the execution of the program.

```
1100 REWIND 4
     REWIND 3
     STOP
```

The statements

```
C
C34       DEFINE FORMATS
C
 1110 FORMAT (10I3)
 1120 FORMAT (26H1      SYSTEM CONFIGURATION)
 1130 FORMAT (1HO,8X,9HEQUIPMENT,25X,4HMTBF,7X,7HTHDM(1),6X,7HTHDM(2),7X
     1,7HTHDM(3),8X,7HTHDM(4)/56X,3HG D,11X,1HN,12X,1HB,15X,1HT//1X)
 1140 FORMAT (I3,1X,8A4,F10.2/4E12.5,2X,2F6.2,2X,A2)
 1150 FORMAT (1H ,1X,I3,2X,8A4,F10.2,2X,4(E12.5,2X))
 1160 FORMAT (50HO  TOO MANY INPUTS OR M IS OUTSIDE ALLOWABLE RANGE)
 1170 FORMAT (I3,1X,F6.2,1X,I3,1X,7A4,2X,L1)
 1180 FORMAT (1HO,5X,9HEVENT NO.,I2,1X,3HIS ,3A4/5X,15HEVENT OCCURRED ,F
     16.2,20H HOURS AFTER TAKEOFF/5X,21HEVENT DESCRIPTION IS ,7A4 /5X,L1
     2)
 1190 FORMAT (1HO,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5,2X,7HPK(J) =,E12.5)
 1200 FORMAT (1HO,10X,40HEFFECTIVENESS FOR OFFENSIVE EVENT NUMBER,I3,1X,
     12HIS,E12.5)
 1210 FORMAT (1HO,10X,36HCUMULATIVE MISSION EFFECTIVENESS IS ,E12.5)
 1220 FORMAT (I3,2X,6E12.5/4(5X,6E12.5))
 1230 FORMAT (23H1      EVENT DESCRIPTION)
 1240 FORMAT (I3)
 1250 FORMAT (1HO //5X,8HRM ARRAY/(14X,4(E12.5,2X)))
 1260 FORMAT (3E12.5)
 1270 FORMAT (1HO,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5,2X,8HCEP(J) =,E12.5,
     12X,7HFA(J) =,E12.5)
 1275 FORMAT( 1HO,10X,3HJ =,I3,2X,9HPCAP(J) =,E12.5)
 1280 FORMAT (25I3)
 1290 FORMAT (2I3,10F7.0)
 1300 FORMAT (1HO,2X,31HCOMPONENT PROBABILITIES OF KILL/
     1 17H  COMP# ELEV#  R=,10(F8.0,1X))
 1310 FORMAT (8E10.4)
 1320 FORMAT (2I6,2(4X,E10.4))
      END
```

define the input and output formats.

Subroutine TWODNZ

The purpose of this routine is to obtain the value of an unknown which lies along a given curve by the linear interpolation method.

The statements

```
      SUBROUTINE TWODNZ (XG,NIZ,Z1,NX,NZ,ANS1)
C
C1        TWO DIMENSIONAL LINEAR INTERPOLATION ROUTINE
C
      DIMENSION Z1(1), ANS1(1)
```

are used to pass information in and out of the subroutine and allow storage for the arrays by use of the DIMENSION statement. The arguments in the calling sequence are identified as follows:

XB = Given value of independent variable

NIZ = Column in which independent variable is stored

Z1 = Name of array in which the variables are stored. Although the array is actually a one-dimensional array, it can be considered to consist of two columns, one for the independent variable and one column for each dependent variable.

NX = Number of values in any column (same for all dependent variables)

NZ = Number of different variables (including the independent variable)

ANS1 = Answer vector

NOTE: Z1 is stored in the following fashion:
Z1(1), Z1(2). . . .Z1(NX), Z2(1). . . .Z2(NX)- -ZNZ(1). . . .ZNZ(NX).
Z1 is stored in increasing order as a single vector with a one-dimensional subscript.

The statements

```
      IXL=(NIZ-1)*NX+1
C2        IF GIVEN VALUE IS LESS THAN LOWER LIMITS OF TABLE, SET RESULTS
C                     EQUAL TO LOWER LIMIT
      IF (XG-Z1(IXL)) 70,70,10
   10 LL=IXL+1
      LU=IXL+NX-1
C3        SEARCH FOR INTERVAL  IN WHICH GIVEN VALUE LIES
      DO 20 J=LL,LU
      IF (XG-Z1(J)) 30,80,20
   20 CONTINUE
C4        IF GIVEN VALUE IS GREATER THAN UPPER LIMIT OF TABLE, SET
C                    RESULTS EQUAL TO UPPER LIMIT
      J=IXL+NX-1
      GO TO 80
```

```
C 5         GIVEN VALUE WITHIN TABLE, CALCULATE INTERPOLATION FACTOR
   30 RAT=(Z1(J)-XG)/(Z1(J)-Z1(J-1))
   40 JP=J-(NIZ-1)*NX
C 6         DO LINEAR INTERPOLATION FOR RESULTS
      DO 60 K=1,NZ
C 7         CHECK FOR ZERO SUBSCRIPT
      IF (JP.EQ.1) GO TO 50
      ANS1(K)=Z1(JP)-RAT*(Z1(JP)-Z1(JP-1))
      GO TO 60
   50 ANS1(1)=Z1(1)
   60 JP=JP+NX
      RETURN
   70 J=IXL
   80 RAT=0.0
      GO TO 40
      END
```

perform the interpolation using a ratio factor. Assuming that the values of the independent variable are called V_i, $i = 1, \ldots n$, the value of V_r corresponding to a given value, T_g is found by the following method:

$$R = \frac{t_g - t_j}{T_{j+1} - T_j}$$

where

$$T_j \leq T_g < T_{j+1}$$

$$V_r = R\,[V_{j+1} - V_j] + V_j$$

There is no extrapolation beyond the values given in the table. When the given value is less than the smallest value of the independent variable in the table, this smallest value is used; similarly when the given value is greater than the largest value of the independent variable in the table, the largest value is used.

USER INFORMATION

Machine Requirements

MISDEM was written for use with the IBM 370/168 computer. This FORTRAN program contains three external references (EXP, MAXO, and AMAX1), all of which are basic ASCII routines. Since MISDEM is almost totally self-contained, it could easily be converted for use on other machines with some minor programming changes. Program 2 is the larger, and occupies approximately 168K bytes of storage to execute as currently dimensioned. Running time for the test (verification) cases required less than one minute for both the electronics and vehicle modes combined.

These specifications are for running the MISDEM model at an IBM 370/168 facility. However, the exact requirements are both machine and facility dependent and should be verified before running the MISDEM model at a specific facility.

Conversion to CDC Machines

Conversion to a CDC machine is exceedingly simple. In program one, line 4,

```
replace  INTEGER* 2  IJN(256)
with     DIMENSION   IJN(256)
```

Between lines 70 and 71 the read statement must be changed to read as follows:

```
READ (5, 1Ø1Ø) I EVENT, T2, NC, (WEAPN(I), I=1, 7), MDT IF (EOF(5)) 840, 711
CONTINUE
```

The changes necessary in program 2 are very similar, starting with the first dimension statement:

```
replace  INTEGER* 2  IJN(256)
with     DIMENSION   IJN(256)
```

Between lines 85 and 90 the read statement must be changed to:

```
READ (5, 1170) I EVENT, TC, NC, (WEAPN(I), I=1, 7), MDT IF (EOF(5)) 1100, 87
CONTINUE
```

Uncontrolled Errors

Machine limitations are probably not significant in most applications. As currently dimensioned, six decimal places have been allowed for the output probabilities. Input data are, at present, probably resulting in inaccurate output beyond the third decimal place. As long as the machine precision exceeds the output parameter precision, there is no ambiguity or error in the output caused by the machine. Uncontrolled mathematical errors occur only in the:

1. Approximation in the reliability algorithms
2. Approximation in the multiple nuclear warhead zone miss probability algorithm
3. Assumed unity kill probability of a nuclear warhead within a lethal radius
4. Assumption of subsystem failure independence, as discussed previously
5. Assumption of failure independence between the vehicle and the electronics.

Controlled Errors Versus Running Time

USE OF NZT. In the conventional electronics mode, controllable errors will occur as a result of suppressing states having (acceptably) low probability. This is accomplished by setting NZT < M where NZT is the maximum number of allowable zeros (failures) in the state (binary -M) vector, and M is the number of subsystems. Because the states having a large number of failures will normally be assigned low probability, their deletion will not significantly affect the output.

This is not true in the nuclear electronics case. A significant amount of probability mass will be allocated to states having several vulnerable subsystems failed. This is due to the tendency of a user to make the damage thresholds common for several subsystems, due to lack of contrary information. When the nuclear threat is significant, it is because failures occur. Then the probabilities assigned to group failures are high due to the unity correlation of failures in the group having the same lethal radius. So the larger the size of such groups, the larger must be the value of NZT to ensure accounting for significant probability mass.

The use of NZT < M is a way to reduce machine use time in electronics modes; therefore, its choice is important. The error caused by NZT < M is difficult to predict, but it can be (in some cases) observed in the output by adding all values of PCAP(J) (mode sequence probability) at a given event. If the sum falls short of 1.0, the defect is due to NZT. The cases where this technique works is where states have not been suppressed by the subsystem criticality (MCR, MPR, MAV) tests (which would otherwise result in an additional loss of probability mass, that would thereby hide the loss caused by NZT alone).

The variable NZT cannot be used to reduce the running time in the vehicle mode because it would introduce large errors as follows. If NZT is less than INCEI (the size of the state vector), the state having all zeros is suppressed. This is the state that causes flight failure in a single epoch, and could carry a significant probability mass if the damage mechanisms are relatively quick.

QUANTIZATION ERRORS. The use of a relatively small number of events to represent a larger number of events in the real world results in time-quantization errors. For the electronics case, the simulated system is not allowed to change its response to encounters (i.e., employ a different mode sequence) except at the specified event time, whereas in the real world, the response could have changed several times in the time interval. The state probability distribution could be in error at the end of the interval for defensive events, as a result of the instability caused by the feedback of countermeasures effectiveness to survivability of countermeasures. When precise results are required, the quantization interval can be decreased to suit. When the mission results are within desired values of the apparent asymptotic values, the interval need be decreased no further. The running time increase is directly proportional to the number of events when it is large.

In the vehicle mode, there is a quantization error due to the fact that the scenario event times may not coincide with the regular intervals required in this mode. One cure for this is the use of a larger number of events, although the impact on running time is much

greater than in the electronics case. The running time in the vehicle case is proportional to $2\ (4\ \log_2\ N)$ for large N, where N is the number of events. Another possible cure for this problem is manipulation of the input data to provide flight time and abort-detection time distribution functions at regular intervals, which are then compatible with MISDEM.

USE OF MCR, MPR AND MAV. The use of critical subsystems (MCR, MPR and MAV) in the electronics case, to reduce running time, suppresses those states having zeros in those subsystems, and results in a loss of probability mass in those modes not requiring such subsystems. If these latter modes are not considered significant to the output, the error is acceptable. The three variables (MCR, MPR and MAV) are identical in function.

Application Notes

DELETION OF SUBSYSTEMS. In the electronics mode, subsystems may be deleted on a mission basis only, by means of the variable ONOFF in the system configuration data. This is not applicable to the vehicle mode, which does not use subsystems in its state vector.

CONTROL OF SUBSYSTEM ON AND OFF TIMES. For purposes of computing reliability in the electronics mode, the program requires a single turn-on time and a single turn-off time for each subsystem. The variables TMN and TMF (for time-on and time-off, respectively) cannot, however, be used to delete subsystems (a function reserved for ONOFF). In fact, it is necessary that the user select TMN and TMF in such a way that the subsystems are "on" for every event whose mode sequence logic requires them "up" for any mode (unless the user is willing to accept the error resulting from the inconsistent inputs).

SOURCES OF DATA. There is no known documentary source of data for the abort-detection time distribution function at present. Some examples of these functions are given to aid the user in deciding how to select input values of this function. Consider first the MISDEM application to involuntary aborts. The latter aborts are defined to be independent of the time the crew detects a need to abort. To model the involuntary aborts, the input abort-detection time distribution functions could be set to zero for all time. The abort then occurs whenever the flight time runs out, not before. Alternatively, it may be assumed that the crew has perfect knowledge of the amount of flight time remaining. In order to model this situation, the abort-detection time distribution functions would be set to 1.0 for all time. The abort then occurs in accordance with the mode sequence logic; e.g., when the flight time remaining becomes less than the normal mission duration.

List of Abbreviations and Symbols
(Simulation Model – Program One)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
AMAX1		IBM utility routine to choose the maximum between two numbers	
BLK		A blank which is used to clear the subsystem names in core	
CN(N)		Ultimately contains the event-type descriptions "offensive" or "defensive"	
CNAME(J,I)		Permanent storage location for the Ith subsystem name	
DEF(N)		A word containing the Holerith character "DEFENSIVE"	
DNAME(J)		Temporary storage location for the subsystem names	
FNAME		Name of the subfunction and/or mode	
HDMT(MPRM)	L(m′,R)	Temporary location for the damage threshold for designated MPRM damage mechanism (gamma "dot", neutron, blast, and thermal)	
IEVENT		Event number	
IJK		Subscript of IJN representing prior state number	
IJN(IJK)		Mode sequence number for the indicated system state	
INCEI		Number of subsystems read-in	
IU	k	Subscript of JU and KU arrays representing subsystem ordinal number	
JCAP		Ultimately the number of mode sequences	
JCOUNT	K	Ultimately the number of system states generated	
JES(L,K,J)		Final binary array defining the subfunction and mode of the Jth mode sequence	
JU(IU,J)		Subsystem ordinal numbers used in the Jth mode sequence	
KS(L,K)		Temporary binary array defining the subfunction/mode number in the current mode sequence	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program One)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
KU(IU)		Temporary binary array defining the ordinal numbers of subsystems used for the current mode sequence	
KW(I)		An array consisting of zeros (0) and ones (1) describing the state of the system	
LCEI		Allowable number of subsystems in equipment configuration	
LCN8		Dimensional constant. Allowable number of equipment name segments	
LE		Number of subfunctions in the current event	
LEE(I)		Number of modes in subfunction I	
LEQ		Synonym for LEE, used as do-loop limit	
LF		Subfunction ordinal number	
L1		Maximum number of modes (all subfunctions)	
L40		Dimensional constant. Maximum number of subfunctions	
L72		Dimensional constant. Maximum number of mode sequences	
LLF(K)		Mode ordinal number for Kth subfunction	
LLQ(I,K)		Next mode number, if current subfunction and mode can be accomplished	
LLQS		Designated value for the next mode in the mode sequence definition	
LL1		Location of current state zero no. 1	
LL2		Location of current state zero no. 2	
LL3		Location of current state zero no. 3	
LL4		Location of current state zero no. 4	
LL5		Location of current state zero no. 5	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program One)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
LL6		Location of current state zero no. 6	
LL7		Location of current state zero no. 7	
LL8		Location of current state zero no. 8	
LMA(I,K,L)		The binary array defining the subsystem numbers (L) required for subfunction I and mode K	
LMAT		Ordinal numbers of subsystems needed for the I,J subfunction/mode input to show location of a (1) in the LMA array	
LMATT(L)		Temporary storage for LMA	
LQ		Next subfunction number if current subfunction and mode equipment requirements are met	
LQSVL		Designated value for the next subfunction in the mode sequence definition	
LTEST		Flow control flag for blocking the selection of the next subfunction if the equipment requirements are not met	
MAV		Ordinal number of critical subsystem (one of three)	
MCR		Ordinal number of critical subsystem	
MD(I,J)		Required mission condition for subfunction and mode	
MDT		Mission descriptor (T or F) specifying mission conditions (cloud cover, visibility, etc.) for the event	
MI(MM)		Flag for each subsystem, MM, set to one or zero depending on ONOFF	
MM		Subscript of MI representing subsystem ordinal number	
MMQ(I,J)		Next mode number if current subfunction and mode do not meet requirement of MDT	
MPR		Ordinal number of critical subsystem	
MPRM	m'	Subscript for HDMT representing the number of a damage mechanism	

List of Abbreviations and Symbols (contd.)

(Simulation Model – Program One)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
MQ(I,K)		Next subfunction number if current subfunction and mode does not meet requirements of the "mission descriptor" MDT	
NC		A flag which identifies both whether the current event type is offensive, defensive-nuclear (also identifies the number of different damage mechanisms), defensive-conventional, or vehicle mode	
NZ		Current number of zeros (0) in the KW array	
NZT		Maximum number of zeros (0) allowed in the KW array (the maximum allowable by the program is 8)	
OFF		A word containing the Holerith characters "OFFENSIVE"	
ONOFF		A flag used to include or exclude a subsystem from the system	
OUT		Mask word against which the input variable ONOFF is tested	
TBFM	MTBF	Mean time between failures	
T2		Time of current event	
WEAPN(7)		Description of weapon used, whether event is offensive or defensive	

List of Abbreviations and Symbols

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
CEP(J)	CEP	Threat circular error probable for Jth mode sequence	feet
CN(3)		Contains the holerith characters for the words OFFENSIVE or DEFENSIVE used in event output description	
CURVE(I,J)	$L(M',r)$	Damage level at radius J for I-type damage mechanism	
DEF(N)		Contains the holerith characters for the word DEFENSIVE	
DELTAT	Δt	Elapsed subsystem time during transition used in reliability calculation	hours
DENAME(8)		Alphanumeric subsystem name characters	
DENUM		Denominator of exponent in calculation of survivability calculation	feet^2
DEN175		Test value for miss distance used to prevent underflow in zone miss probability calculation	
ET	$E_T(N)$	Expected number of targets killed, cumulative	
EXM		Temporary value for second term in nuclear zone miss probability	
EX1		Temporary value for first term in q′ calculation and in conventional miss probability calculation	
EX2		Second term in conventional miss probability calculation	
EXM1		First term in nuclear zone miss probability calculation	
FA(J)	SPd	Expected number of weapons arriving in target vicinity for Jth mode sequence	
FTDNN(I,J)		Probability of abort-detection before time J, given flight time equal to J	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
FTFNN(I)		Probability of flight time less than I	
GRAPH(IJ)		Damage threshold vs. nuclear weapon mis data being read into interpolation subroutine	
HDMT(1)		Gamma dot damage threshold	rads/sec
HDMT(2)		Neutron damage threshold	neutrons/cm^2
HDMT(3)		Blast damage threshold	lb/in^2
HDMT(4)		Thermal damage threshold	calories/in^2
ICOUNT	i	Prior event state number	
IEVENT		Prior event number (normal mission)	
II		Prior event type: II = 0 offensive 1-5 defensive-nuclear 6 defensive quick conventional kills 7 defensive slow kill	
II1		Location of prior state zero no. 1	
II2		Location of prior state zero no. 2	
II3		Location of prior state zero no. 3	
II4		Location of prior state zero no. 4	
II5		Location of prior state zero no. 5	
II6		Location of prior state zero no. 6	
II7		Location of prior state zero no. 7	
II8		Location of prior state zero no. 8	
IINCEI		One-half the number of elements in the state vector (for the vehicle mode only)	
IJK	i	Prior state number	
IJN(IJK)	J(i)	Mode sequence number for prior state number IJK	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
INCEI		Ultimately, the number of subsystems read from input data	
INE		One more than INCEI	
INIT		"Next" subsystem number, to be tested for lethal radius against "this" subsystem number	
IS		Temporary value of current damage mechanism type	
ISUB(I)		Temporary damage mechanism type number for Ith damage mechanism	
IT(M)		Prior state binary bit (before reordering) for Mth subsystem	
ITDN	τ_i	Prior event abort-detection time, referenced to current event time	
ITFN	t_i	Prior event flight time, referenced to current event time	
ITORE		Temporary value of IZ(L)	
ITT(M)		Reordered prior state binary bit for subsystem M	
ITTD		Prior event abort-detection time, referenced to prior event time	
IV(L)		Prior event vulnerability index number for Lth subsystem	
IZ(L)	IZ(k)	Current event vulnerability index number for subsystem L	
JCAP		Number of mode sequence read from IJN tape	
JCOUNT		Current event state number	
JEVENT		Event number in normal mission used to initialize state probability for an abort mission	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
JINCEI		One plus one-half the number of elements in the state vector (for the vehicle mode only)	
JV		Temporary value for IV(M)	
KMAX		Number of different offset zones	
KMAX1		One less than the number of offset weapon zone radii (conventional weapons)	
KW(M)		Current state binary bit for subsystem M	
KWW(I)		Reordered current state binary bit for subsystem I	
L72		Maximum number of mode sequences generated by program one for any event (40)	
LCEI		Allowable number of subsystems in the equipment configuration	
LCN8		Allowable number of equipment name segments	
LL1		Location of current state zero no. 1	
LL2		Location of current state zero no. 2	
LL3		Location of current state zero no. 3	
LL4		Location of current state zero no. 4	
LL5		Location of current state zero no. 5	
LL6		Location of current state zero no. 6	
LL7		Location of current state zero no. 7	
LL8		Location of current state zero no. 8	
LMAT(L)		Ordinal number of subsystem required (electronics mode) or location of "one" in state vector (vehicle mode)	
LMAX		Number of different elevation angles associated with warhead offset trajectory	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
LOO		"This" subsystem number used in defining vulnerability index numbers	
M	K	Subsystem ordinal number	
MAV		Ordinal number of critical subsystem	
MCR		Ordinal number of critical subsystem	
MDT		Mission descriptor: T = scenario constraint imposed F = scenario constraint not imposed	
MI(M)		On/off flag for subsystem M	
MLTH	L_m	Mission length measured from initiation of first threat exposure	
MPR		Ordinal number of critical subsystem	
MPRM	m'	Damage mechanism type number	
MTBF(M)	MTBF	Mean time between failures	hours
NABORT		Number of the event at which the abort path is initiated	
NC		A flag: NC = 0 offensive event NC = 1, 2, 3, 4, or 5 nuclear defensive event with NC damage mechanisms NC = 6 conventional defensive event NC = 7 vehicle simulation event	
NF		A flag: NF = 0 nuclear damage is possible NF = 1 nuclear damage is not possible	
NPOINT		Number of points in the miss distance vs. damage level table lookup	
NTDN	τ_j	Current event abort-detection time, referenced to current event time	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
NTFN	t_j	Current event flight time, referenced to current event time	
NTTD		Current event abort-detection time, referenced to prior event time	
NTTF		Current event flight time, referenced to prior event time	
NZ		A counter to keep track of current number of zeros in prior state vector	
NZI		A counter to keep track of current number of zeros in current state vector	
NZT		Maximum number of zeros allowed in state vector for the mission	
N2		Two times the number of lethal radii	feet
OFF(N)		Contains the Holerith characters for the word OFFENSIVE	
ONOFF		Flag used to include or exclude a subsystem from the system (T = include, F = exclude)	
OUT		Variable against which input variable ONOFF is tested	
PARIVE		Probability of arrival	
PCAP(J)		Probability of Jth mode sequence	
PCKILL(M,L,K)	k	Probability of kill of subsystem M given offset zone K and elevation L (conventional)	
PCSURV(M,L,K)	v_k	Probability of survival of subsystem M (conventional)	
PI(IJK)	P_i	Probability for prior state no. IJK	
PJ(L)	$P_{j,n}$	Probability for current state no. L	
PK(J)	$P_{k,j}$	Weapon delivery kill probability for Jth mode sequence	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
PM(M)	d_k	Probability of reliable operation in transition (before reordering)	
PMISS(K,IJK)	$P_{miss}(k)$	Probability of threat miss distance in zone K, given state IJK	
PMM(M)	d_k	Reliability of each subsystem M in transition (after reordering)	
PTDN(I,J)		Probability that abort-detection time is J, given that flight time is I	
PTFN(I)		Probability that flight time is I	
QO(IJK)	q'	The probability of nuclear weapon miss distance exceeding the maximum lethal radius	
QPRM(M,IJK)	q'_k	Probability of nuclear weapon miss in zone associated with Mth subsystem, given the IJKth state	
R(K)	R_k	Inner radius of the Kth offset zone	feet
RA(M)		Temporary value of subsystem lethal radius obtained from linear interpolation routine	feet
RESUL(2)	r_k	Lethal radius output of interpolation subroutine	feet
RM2	r_k^2	Square of the radius RM(1) or R(k)	$feet^2$
RMM12		Square radius associated with second term in nuclear zone miss probability calculation	$feet^2$
RMM2		Square of the radius RM(M)	$feet^2$
SIGMA	σ	Standard deviation of miss distance whether nuclear or convention weapon	feet
STORE	r_k	Temporary value of RM(k)	feet
SUM	$P_{k,n}$	Average (over all modes) target kill probability in nth event	

List of Abbreviations and Symbols (contd)

(Simulation Model – Program Two)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
SUMPTD		Ultimately, the probability of abort-detection time (due to an encounter) greater than or equal to the current abort detection time	
SUMTRA		Ultimately, the sum of all transition probabilities resulting from all LMAX offset trajectories for one zone	
SUMTRK		Ultimately, the sum of all transition probabilities resulting from all KMAX offset trajectories	
T1		Time of prior event	hours
T2		Time of current event	hours
TBFM	MTBF	Mean time between failures	hours
THDM(M,IS)		Temporary value of damage threshold for subsystem M, damage type IS	(See HDMT)
TIMEF(M)		Time that subsystem M is turned off	hours
TIMEN(M)		Time that subsystem M is turned on	hours
TMF		Time that subsystem is turned off	hours
TMN		Time that subsystem is turned on	hours
TRANS	$P_{j/i,n}$	Transition probability for state pair under consideration	
TRANS1		Temporary value of TRANS in the vehicle transition algorithm (slow threat damage)	
TWODNZ		Name of subroutine for two-dimensional interpolation	
WEAPN(7)		Description of weapon whether offensive or threat	

List of Abbreviations and Symbols

(Simulation Model – Subroutine TWODNZ)

Abbreviation or symbol	Equivalent in mathematical model	Definition	Units
J		Search subscript	
K		Subscript used to fill answer vector	
JP		Current subscript of upper value of result interval	
LL		Subscript of lower value in result interval	
LU		Subscript of upper value in result interval	
NX		Number of values in any column (same for all parameters)	
NZ		Number of different parameters	
XG		Given value of independent variable	
Z1		Name of table to perform lookup on	
IXL		Subscript of first value in independent variable column	
NIZ		Column of independent variable	
RAT		Ratio factor for linear interpolation	
ANS1		Answer vector	
TWODNZ		Subroutine name and acronym for two-dimensional linear interpolation	

DISTRIBUTION LIST

Aeronautical Systems Division (AFSC)
Wright-Patterson AFB, OH 45433
Attn: ASD/ACCX (MAJ F. Munguia)
Attn: ASD/ENESS (P. T. Marth)
Attn: ASD/ENFTV (D. J. Wallick) (2 copies)
Attn: ASD/XROL (F. Campanile)
Attn: ASD/XROL (R. K. Frick)
Attn: ASD/XROT (G. B. Bennett)
Attn: ASD/XRU (S. E. Tate)
Attn: ASD/YPEF (C. Gebhard)

Aerospace Medical Research Laboratories
AMRL/MEA
Area B, Bldg. 33
Wright-Patterson AFB, OH 45433
Attn: AMRL/MEA (CAPT G. J. Valentino)

Aerospace Rescue and Recovery Service (MAC)
Scott AFB, IL 62225
Attn: ARRS/DOQ (CAPT T. J. Doherty)

Air Force Acquisition Logistics Division
Wright-Patterson AFB, OH 45433
Attn: AFALD/PTEA (MAJ D. Waltman)

Air Force Avionics Laboratory
Wright-Patterson AFB, OH 45433
Attn: AFAL/WRA-1 (E. Leaphart)
Attn: AFAL/WRP (W. F. Bahret)

Air Force Flight Dynamics Laboratory
Wright-Patterson AFB, OH 45433
Attn: AFFDL/FER (C. V. Mayrand)
Attn: AFFDL/FES (CDIC) (2 copies)
Attn: AFFDL/FES (J. Hodges)
Attn: AFFDL/FES (R. W. Lauzze)
Attn: AFFDL/FES (D. W. Voyls)
Attn: AFFDL/FGL (F. R. Taylor)
Attn: AFFDL/TST (Library)

Air Force Logistic Command
Wright-Patterson AFB, OH 45433
Attn: AFLC/LOE (Commander)

Air Force Systems Command
Andrews AFB, DC 20334
Attn: AFSC/DLCAA (P. L. Sandler)

Air Force Test and Evaluation Center
Kirtland AFB, NM 87115
Attn: AFTEC/OAR (K. Campbell)

Air Force Weapons Laboratory
Kirtland AFB, NM 87117
Attn: AFWL/PGV (MAJ H. Rede)

Applied Technology Laboratory
Army Research & Technology Laboratory (AVRADCOM)
Ft. Eustis, VA 23604
Attn: DAVDL-EU-MOS (Mr. Merritt)
Attn: DAVDL-EU-MOS (S. Pociluyko)
Attn: DAVDL-EU-MOS (C. M. Pedriani)
Attn: DAVDL-EU-MOS (J. T. Robinson)

Armament Development and Test Center
Eglin AFB, FL 32542
Attn: ADTC/DLODL (Technical Library)
Attn: ADTC/XRSP (E. Blair)

Army Armament Research and Development Command
Dover, NJ 07801
Attn: DRDAR-LCS (S. K. Einbinder)

Army Aviation Research & Development Command
P.O. Box 209
St. Louis, MO 63166
Attn: DRCPM-ASE-TM (MAJ Schwend) (2 copies)

Army Ballistic Research Laboratories
Aberdeen Proving Ground, MD 21005
Attn: DRXBR-VL (D. W. Mowrer)

Army Foreign Science and Technology Center
220 Seventh St., NE
Charlottesville, VA 22901
Attn: DRXST-BA3 (E. R. McInturff)

Army Materials and Mechanics Research Center
Watertown, MA 02172
Attn: DRXMR-EM (A. A. Anctil)
Attn: DRXMR-MI (C. F. Hickey, Jr.)
Attn: DRXMR-PL (M. M. Murphy) (2 copies)
Attn: DRXMR-RD (R. W. Lewis)

Army Materiel Systems Analysis Activity
Aberdeen Proving Ground, MD 21005
Attn: DRXSY-AA (Director)
Attn: DRXSY-AD (H. X. Peaker)
Attn: DRXSY-J
Attn: DRXSY-S (J. R. Lindenmuth)

Chief of Naval Operations, Office of the
Room 5C735, Pentagon
Washington D.C. 20350
Attn: OPNAV-982E3L (LT COL W. A. Allanson)

David W. Taylor Naval Ship R&D Center
Bethesda, MD 20084
Attn: Code 1740.2 (F. J. Fisch)
Attn: Code 522 (Commander) (4 copies)

Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209
Attn: S. Zakanycz

Defense Documentation Center
Cameron Station, Bldg. 5
Alexandria, VA 22314
Attn: DDC-TCA (12 copies)

Defense Systems Management College
Ft. Belvoir, VA 22060
Attn: W. Schmidt

Department of Transportation - FAA
2100 Second St., SW, Rm 1400C
Washington, DC 20591
Attn: ARD-520 (R. A. Kirsch)

Deputy Chief of Staff (AIR)
Marine Corps Headquarters
Washington, DC 20380
Attn: ASA-21 (LT COL F. C. Regan)

ERADCOM
Fort Monmouth, NJ 07703
Attn: DAVAA-I (S. Zywotow, Avionics R&D Activity)
Attn: DELEW-P (M. Adler, EW Protection Division)
Attn: DELSD-EM (C. Goldy, Electronics R&D Command)

Foreign Technology Division (AFSC)
Wright-Patterson AFB, OH 45433
Attn: FTD/SDED (C. W. Caudy)
Attn: FTD/SDNS-3 (LT Saylor/73041)

HQ Air Logistics Command
McClellan AFB, CA 95652
Attn: SM/MMSRBC (D. E. Snider)

HQ SAC
Offutt AFB, NB 68113
Attn: JSTPS/JPTB (MAJ C. O. Cox)
Attn: NRI/STINFO Library

Marine Corps Development Center
Quantico, VA 22134
Attn: D-091 (LT COL J. Givan)

NASA - Ames Research Center
Army Air Mobility R&D Laboratory
Mail Stop 207-5
Moffett Field, CA 94035
Attn: DAVDL-AS (V. L. J. Di Rito)

NASA - Johnson Spacecraft Center
Houston, TX 77058
Attn: EC (F. S. Dawn)

NASA - Lewis Research Center
21000 Brookpark Rd.
Mail Stop 500-202
Cleveland, OH 44135
Attn: Library (D. Morris)

National Bureau of Standards
Building 225, Rm A62
Washington, DC 20234
Attn: I. A. Benjamin

Naval Air Development Center
Warminster, PA 18974
Attn: Code 097 (MAJ W. Boeck)
Attn: Code 701 (B. Vafakos)
Attn: Code 2012 (M. C. Mitchell)
Attn: Code 2012 (R. H. Beliveau)
Attn: Code 6083 (S. L. Huang)
Attn: Code 6099 (R. A. Ritter)

Naval Air Propulsion Test Center
P.O. Box 7176
Trenton, NJ 08628
Attn: PE3 (D. Wysocki)
Attn: PE42 (R. W. Vizzinni)

Naval Air Systems
Airtevron One
Patuxent River MD 20653
Attn: LT R. N. Freedman

Naval Air Systems Command
Washington, DC 20361
Attn: AIR-03PA4 (T. S. Momiyama)
Attn: AIR-330B (E. A. Lichtman)
Attn: AIR-350 (E. M. Fisher)
Attn: AIR-503W1 (E. A. Thibault)
Attn: AIR-52014 (L. Sztan)
Attn: AIR-5204A (D. Atkinson) (2 copies)
Attn: AIR-5204J (D. P. Bartz)
Attn: AIR-530121A (A. Besaha)
Attn: AIR-5303
Attn: AIR-530313 (R. D. Hume)
Attn: AIR-53051A (P. Kicos)
Attn: AIR-5323A
Attn: AIR-5323K (K. Chang)
Attn: AIR-5323Z (S. Englander)
Attn: AIR-53603A (G. W. Gigioli)
Attn: AIR-620B1 (LCDR K. K. Miles)
Attn: AIR-954 (Tech. Library)
Attn: PMA-2692A1 (R. W. Wills)
Attn: PMA-2694 (T. S. Meek)

Naval Ordnance Station
Indian Head, MD 20640
Attn: Code 5123F (D. H. Brooks)

Naval Postgraduate School
Monterey, CA 93940
Attn: Code 67BP (R. E. Ball)
Attn: Library

Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375
Attn: Code 5730 (E. E. Koos)

Naval Surface Weapons Center
Dahlgren Laboratory
Dahlgren, VA 22448
Attn: CK-2301 (J. E. Mitchell)
Attn: CN-61 (J. S. Nerrie)
Attn: DG-10 (J. E. Ball)
Attn: G-11 (F. J. Petranka)
Attn: Library

Naval Surface Weapons Center
White Oak Laboratory
Silver Spring, MD 20910
Attn: CN-13

Naval Weapons Center
China Lake, CA 93555
Attn: Code 3181 (C. Padgett) (2 copies)

Naval Weapons Engineering Support Activity
Systems Analysis Dept.
Bldg. 210-2 (ESA-19)
Washington Navy Yard
Washington D.C. 20374
Attn: Code ESA-1923 (C. W. Stokes III) (2 copies)
Attn: Code 11621 (J. Stasko)

Naval Weapons Support Center
Crane, IN 47522
Attn: Code 502 (N. L. Papke)

Operational Test & Evaluation Force
Norfolk, VA 23511
Attn: NOB (T. H. McCants, Science Advisor)

Pacific Missile Test Center
Point Mugu, CA 93042
Attn: Code 1151 (R. L. Nielson)
Attn: Code 4253-3 (Naval Air Station, Technical Library) (2 copies)

Warner Robins Air Logistics Center
Robins AFB, GA 31098
Attn: WRALC/MMETE (LT W. Shelton)

Aerojet ElectroSystems Company
A Div. of Aerojet-General Corp.
1100 W. Hollyvale Street
Azusa, CA 91702
Attn: A. R. Moorman

AiResearch Manufacturing Co. of Arizona
A Division of the Garrett Corp.
P.O. Box 5217
Phoenix, AZ 85010
Attn: G. L. Merrill

Armament Systems, Inc.
712-F North Valley Street
Anaheim, CA 92801
Attn: J. Musch

A. T. Kearney and Company, Inc.
100 South Wacker Drive
Chicago, IL 60606
Attn: R. H. Rose

AVCO
Lycoming Division
550 So. Main St.
Stratford, CT 06497
Attn: R. Cuny

Battelle Memorial Institute
505 King Ave.
Columbus, OH 43201
Attn: J. H. Brown, Jr.

Bell Helicopter Textron
Division of Textron Inc.
P.O. Box 482
Fort Worth, TX 76101
Attn: Security/Dept. 12, J. R. Johnson

The Boeing Aerospace Company
P.O. Box 3999
Seattle, WA 98124
Attn: J. G. Avery, M/S 4C-08
Attn: R. G. Blaisdell, M/S 8W-41
Attn: R. J. Helzer, M/S 47-28

The Boeing Company
Vertol Division
Boeing Center
P.O. Box 16858
Philadelphia, PA 19142
Attn: J. E. Gonsalves, M/S P32-19 (2 copies)

The Boeing Company
Wichita Division
3801 S. Oliver St.
Wichita, KS 67210
Attn: H. E. Corner, M/S K16-67
Attn: D. Y. Sink, M/S K16-14

Calspan Corp.
P.O. Box 235
Buffalo, NY 14221
Attn: Library (V. M. Young)

COMARCO inc
1417 N. Norma
Ridgecrest, CA 93555
Attn: G. Russell (2 copies)

Fairchild Industries, Inc.
Fairchild Republic Co.
Conklin Street
Farmingdale, L.I., NY 11735
Attn: J. A. Arrighi
Attn: G. Mott
Attn: Engineering Library (G. A. Mauter)

Falcon Research and Development Co.
2350 Alamo Ave., SE
Albuquerque, NM 87106
Attn: W. L. Baker

Falcon Research and Development Co.
696 Fairmount Ave.
Baltimore, MD 21204
Attn: J. A. Silva

Ford Aerospace and Communications Corp.
Ford Road, P.O. Box A
Newport Beach, CA 92663
Attn: Library

General Dynamics Corp.
Fort Worth Division
Grants Lane, P.O. Box 748
Fort Worth, TX 76101
Attn: P. R. deTonnancour/G. W. Bowen

General Electric Co.
Aircraft Engine Group
1000 Western Ave.
West Lynn, MA 01910
Attn: J. M. Wannemacher

General Electric Co.
Aircraft Engine Business Group
Evendale Plant
Cincinnati, OH 45215
Attn: AEG Technical Information Center (J. J. Brady)

General Research Corporation
P.O. Box 6770
5383 Hollister Avenue
Santa Barbara, CA 93111
Attn: J. H. Cunningham

General Research Corporation
SWL Division
Suite 700, Park Place
7926 Jones Branch Dr.
Mclean, VA 22101
Attn: T. King

Goodyear Aerospace Corp.
1210 Massillon Rd.
Akron, OH 44315
Attn: J. E. Wells, D/959G
Attn: Library, D/152G (R. L. Vittitoe/J. R. Wolfersberger) (3 copies)

Grumman Aerospace Corp.
South Oyster Bay Rd.
Bethpage, NY 11714
Attn: J. P. Archey Jr., Dept. 662, Mail C42-05
Attn: R. W. Harvey, Mail C27-05
Attn: H. L. Henze, B16-25
Attn: J. Noack, M/S B10-25
Attn: Technical Information Center, Plant 35 L01-35 (H. B. Smith)

Hughes Helicopters
A Division of Summa Corp.
Centinela Ave. & Teale St.
Culver City, CA 90230
Attn: Library, 2/T2124 (D. K. Goss)

Institute for Defense Analyses
400 Army - Navy Drive
Arlington, VA 22202
Attn: Technical Information Center, DIMO (P. Okamoto)

171

IIT Research Institute
10 West 35 Street
Chicago, IL 60616
Attn: I. Pincus

The Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20810
Attn: C. F. Meyer

Kaman Aerospace Corporation
Old Windsor Rd.
Bloomfield, CT 06002
Attn: H. E. Showalter

Lockheed-California Co.
A Division of Lockheed Aircraft Corp.
2555 Hollywood Way
P.O. Box 551
Burbank, CA 91520
Attn: Technological Information Center, 84-40 Unit 35, Plant A-1
Attn: G. E. Raymer, D/75-84 Bldg. 63 A-1 (2 copies)

Lockheed-Georgia Co.
A Division of Lockheed Aircraft Corp.
86 S. Cobb Drive
Marietta, GA 30063
Attn: D. R. Scarbrough, 72-08 Zone 12
Attn: Sci-Tech Info Center, 72-34 Zone 26 (T. J. Kopkin)

Lockheed Missiles & Space Company Inc.
A Subsidiary of Lockheed Aircraft Corp.
P.O. Box 504
Sunnyvale, CA 94088
Attn: G. R. Evans (5501-572-5)

Martin Marietta Corp.
Orlando Division
P.O. Box 5837
Orlando, FL 32855
Attn: Library (M. C. Griffith, MP-30)

McDonnell Douglas Corp.
Douglas Aircraft Company
3855 Lakewood Blvd.
Long Beach, CA 90846
Attn: Technical Library, C1-250/36-84 AUTO 14-78 (3 copies)

McDonnell Douglas Corp.
P.O. Box 516
St. Louis, MO 63166
Attn: R. D. Detrich, Dept. 022

New Mexico Institute of Mining and Technology
Campus Station
Socorro, NM 87801
Attn: Tera

Northrop Corp.
Aircraft Division
3901 W. Broadway
Hawthorne, CA 90250
Attn: J. H. Bach, 2130/83
Attn: V. B. Bertagna, 3451/81
Attn: H. W. Jones, 3360/82
Attn: J. F. Paris 3628/83

Northrop Corp.
Ventura Division
1515 Rancho Conejo Blvd.
P.O. Box 2500
Newbury Park, CA 91320
Attn: M. Raine

The Rand Corp.
1700 Main St.
Santa Monica, CA 90406
Attn: N. W. Crawford

R&D Associates
P.O. Box 9695
Marina Del Rey, CA 90291
Attn: B. Jaeger

Rockwell International Corp.
Los Angeles Division
5701 W. Imperial Hwy
Los Angeles, CA 90009
Attn: W. L. Jackson
Attn: S. C. Mellin, AD-25
Attn: R. Moonan, AB78 (2 copies)
Attn: J. H. Howard, AB82

Rockwell International Corp.
4300 E. Fifth Ave.
P.O. Box 1259
Columbus, OH 43216
Attn: Technical Information Center (D. Z. Cox) (2 copies)

Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78284
Attn: P. H. Zabel, Div. 02

Stanford Research Institute
333 Ravenswood Ave.
Menlo Park, CA 94025
Attn: J. Golins

System Planning Corporation
1500 Wilson Blvd., Suite 1300
Arlington, VA 22209
Attn: J. A. Navarro

Teledyne Ryan Aeronautical
2701 Harbor Dr.
San Diego, CA 92112
Attn: Technical Information Services (W. E. Ebner)

Uniroyal Plastic Products
Division of UNIROYAL, Inc.
407 N. Main Street
Mishawaka, IN 46544
Attn: J. D. Galloway

United Technologies Corporation
Pratt & Whitney Aircraft Group
Government Products Division
P.O. Box 2691
West Palm Beach, FL 33402
Attn: P. E. Desrosiers, Security Officer
Attn: J. Fyfe, Mail E-39

Vought Corporation
P.O. Box 5907
Dallas, TX 75222
Attn: D. M. Reedy, 2-30100

ABSTRACT CARD

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

ABSTRACT CARD

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

Naval Weapons Center

MISDEM Computer Simulation – Volume II (Analyst Manual), by G. L. Gallien and S. C. Silver, Rockwell International, Los Angeles, CA, NWC, May 1979, 166 pp. (JTCG/AS-76-S-004, publication UNCLASSIFIED).

The MISDEM (Mission/Damage Effectiveness Model) is a survivability/vulnerability model that transforms aircraft subsystem probabilities of survival into probabilities of aircraft survival and probabilities of various aircraft response modes, such as flight, countermeasures, and weapon delivery modes having different degrees of effectiveness. The model may be used to

(over)
1 card, 8 copies

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.

JTCG/AS-76-S-004

compute measures of effectiveness, such as numbers of targets killed in a mission or a campaign. It is intended for use in measuring the impact of vulnerability of subsystems on aircraft survival and effectiveness for unenhanced or protected subsystems. This analyst manual contains math model, applications, computer code, derivation, and test cases.